

FreeNIBS/STArad

Web interface and authorization, authentication and billing module for
FreeRADIUS server

User Manual

Stanislav Korsakov
STAssoft.net (c) 2001-2008
Yaroslavl

TABLE OF CONTENTS

INTRODUCTION.....	4
What is FreeNIBS/STArad?.....	4
.....	5
Hardware and software requirements.....	5
Installation, peculiarities of various versions and the general development plan.....	6
How to support the development of FreeNIBS/STArad?.....	7
Objects within the module.....	7
FEATURES.....	8
Service scopes.....	9
HOW FreeNIBS OPERATES?.....	11
 Authorization.....	11
 Description of black/white list system operation.....	14
 Accounting.....	16
 Web interface (control interface).....	17
 General NETSHe system concept.....	17
 Getting started.....	18
 Entry of companies, people, providers and other basic information.....	18
 Tariffs for the subscribers.....	26
 Creating and editing subscribers.....	40
 Statistical data and reports.....	46
Annex 1.....	47
Annex 2. Setting of individual parameters.....	48
 Stale sessions.....	48
Annex 3. Description of external subscriber/service management interface.....	50

INTRODUCTION

This User Manual is based on the manual for FreeNIBS v0.0.3 and contains the original text and authors' personal opinions.

This Manual will be updated as far as possible and if new functional properties of FreeNIBS/STArad appear.

This software product includes software of the third parties. The author of this Manual does not undertake any obligations for any update of this document and of the whole FreeNIBS/STArad module.

The author expresses his gratitude to the following people and institutions:

- Sergey Derkach (Neon), the author of the first FreeNIBS versions.
- Pavel Bochkaryov for the help rendered during development and debugging.
- Josep Pocalles for ideological and financial support.
- Soporte Bankoi for financial support.
- EkoPLC for financial support.

What is FreeNIBS/STArad?

FreeNIBS/STArad is the authorization, authentication and billing module for dialin subscribers (subscribers accessing the service by way of PPP/PPPOE/PPTP protocols and through the services requiring/using authorization/accounting via the radius server).

This module operates together with the FreeRADIUS radius server or the radius server from STAssoft.net.

The module operates with MySQL, PgSQL and Oracle databases. Drivers for other relational database management systems (RDBMS) can also be developed.

The module performs subscriber billing on a real-time basis (in prepaid mode - rendering of service after pre-payment, postpaid mode - rendering of service with subsequent payment as well in combined mode).

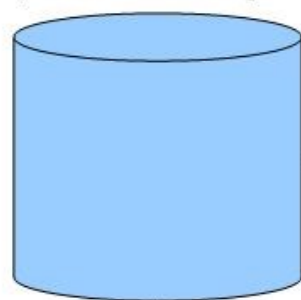
The module has comprehensive tuning capabilities. In particular it allows setting of billing rules and service rendering parameters for subscriber groups, access servers and individual subscribers.

The module operates on all platforms and in all operating systems where FreeRADIUS radius server and STArad radius server are operational.

FreeRADIUS is a radius server for UNIX-like operating systems, which is also distributed as source code under the GPL license. See the official web page <http://www.freeradius.org> for details.

The module as well as the radius server are freeware. The module is distributed as source code under the GPL license ver. 2.

Access server configured for termination of vpn (or) ppp/pppoe/pptp connections. It is also to be configured to be able to send authorization, accounting start, accounting stop and accounting alive requests



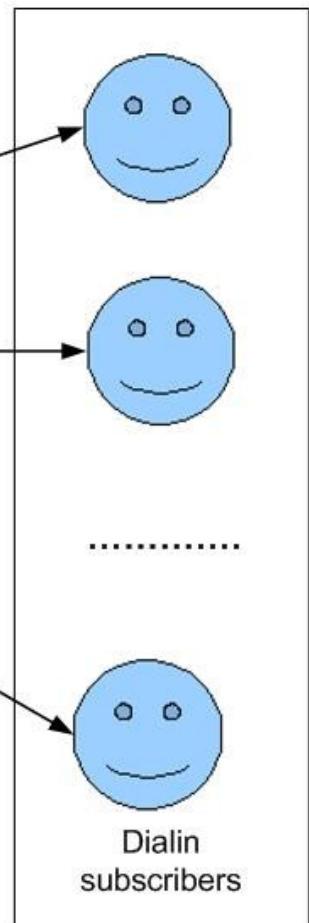
Communication channel

Communication channel



Server

Server with FreeRADIUS and FreeNIBS for management and billing of dialin subscribers



Hardware and software requirements

This software operates under Linux family operating systems (Debian, RedHat and similar OS's, Gentoo, etc.), FreeBSD, NetBSD, OpenBSD. There are no major problems preventing this software to operate under Solaris and MacOS. This software has not been run under Windows family operating systems.

This software requires an installed and tuned RDBMS server, such as MySQL, PostgreSQL or Oracle (version 3.0.0 uses PostgreSQL as the main data storage).

This software product requires an installed Perl language interpreter and version 4 or 5 PHP language interpreter (for version 5 it is recommended to use the relevant versions of TCPDF and JpGraph modules).

The computer is to be also provided with the relevant development tools in order to install and assemble the software:

- Compiler from the language C
- automake
- m4

The system also requires a web server for the web interface (apache, lighthttpd). Version 3.0.0 uses and integrated nanoWeb web server.

At present the versions are numbered as follows:

Numbering 1.x.x is used instead of 0.1.x.

Numbering 2.x.x is used instead of 0.2.x.

Development of the versions 1.x and 2.x is terminated.

Support of the versions 1.x is terminated.

Version 3 of the FreeNIBS module and version 4 of STARad are being developed at the moment. See the official web page of the project <http://stasoft.net/freenibs> for details.

Installation, peculiarities of various versions and the general development plan

FreeNIBS/STARad is a popular product in ex-USSR countries, which has at least several hundreds installations (as per the assessment of the author of this Manual).

Software solutions based on this product are also installed in Europe, USA and India.

FreeNIBS/STARad is widely used by Internet service providers and in Ethernet networks. Besides, FreeNIBS and FreeNIBS based software solutions can be used at the corporate level for the arrangement of controlled access to individual services (e. g. Internet access) and accounting.

With regards to its functional capabilities FreeNIBS supersedes many commercial products.

At the moment 3 branches of FreeNIBS are available.

Versions 1.x.x are totally compatible with the original FreeNIBS (from Neon) with refactored code here and there, eliminated errors and the new (or improved) functional capabilities, which the developers had managed to introduce without changing the data structure. At the moment the development of the versions 1.x.x is terminated. New releases will include only the elimination of possible errors and features from the older versions, which can be ported without changing the data structure.

At the moment any development and support of the versions 1.x.x is terminated. We strongly recommend that the users of the original FreeNIBS and the versions 1.x.x should upgrade to the version 2.x

Versions 2.x.x include the features of the versions 1.x.x and have a slightly modified data structure in order to ensure some new features. Special emphasis was placed upon the subscriber billing on the basis of the traffic with various charges depending on the size of the traffic (the so-called "billing with traffic

gradations”). Besides, this version allows setting of various price lists for various access servers.

At the time the development of the branch 2.x.x versions is completed. Version 2.x.x is a stable and fully functional software solution.

New releases will include only the elimination of possible errors and the features from the older versions, which can be ported without changing the data structure.

Versions 3.x.x include the features of the versions 2.x.x and have totally refactored data structure, which was necessary in order to introduce new features (authorization by mac address, automatic number identifier, etc., postpaid billing) and to ensure normal consistent operation with other modules.

Version 4.x.x is a future-proof and the next stable software branch. It will include all the newest solutions from version 3 and the SQLite backend. Beginning from version 4 the software product will be called STArad.

How to support the development of FreeNIBS/STArad?

We will greatly appreciate your support in the development of FreeNIBS. Your help and support can be as follows:

- Joint development of the module.
- Code compilation and testing at 64 bit architectures.
- Testing and Debugging.
- Data definition and data manipulation for various RDBMS's.
- Writing drivers for other RDBMS's.
- Writing scripts and service utilities.
- Joint development of web interfaces.
- Writing documentation and configuration examples for different access servers.
- Conceptualization of the module. Development and planning of the module's features.
- Translation of the documentation.
- Financial support.

Development group and FreeNIBS users will greatly appreciate your help and support in any form.

Objects within the module

The module operates with a set of objects. Each object has a property set. The properties describe service rendering conditions and conditions of payment for the scope of rendered services (billing rules). Objects in the sense of the module are as follows:

- Radius server(s) is a means of subscriber authorization, accounting and billing.

What is RADIUS?

RADIUS is a protocol consisting of 2 parts: AUTH (authorization) part and ACCT part (accounting). AUTH part receives authorization requests (password

checking) and responds with permission or denial. ACCT part receives data with regards to session start, session end and, what is possible, session in progress.

The packets, corresponding to the protocol, use udp-protocol and go to the ports: each part to its port:

1812 udp (1645 udp) - Radius auth

1813 udp (1646 udp) - Radius acct

Both parts operate independently.

- Access server(s) is a means of subscriber access to the service.
- Subscriber with a unique ID login and password (as a minimum property set).
- Subscriber group (or packet), which defines the general set of properties for several subscribers.
- Price list, which defines the cost of rendered services for each individual packet, access server or subscriber.
- Access time is the time within which the subscriber (all subscribers of the packet) can access the service. It is also the time of communication session.
- Holiday ("special" day) is the day when the cost of rendered services is calculated in accordance with a special price list.
- Day of service rendering is the day of the week within which the subscriber had at least one session.
- Session is the communication session within which the subscriber used the service. Session starts from the moment of successful subscriber authorization at the radius server.
- Service scope is an agreed method of rendered service measurement. It can have various measurement units: duration, traffic, fact of connection or any combination thereof.
- Limit is a rule applied to rendering of service to the subscriber. The utmost limit is the denial (permanent or temporary) to provide the service. Example of a limit is the number of concurrent logins.

FEATURES

- All subscribers and groups (packets) are stored in the database/
- Various databases can be used (MySQL, Postgresql, Oracle, etc.)
- Bills management
- Creation of unlimited number of packets with various configurations
- Operation in traffic, time and traffic + time counting mode
- Counting of any traffic
- Network capability - VPN
- Operation in limited mode with regards to the funds on account, time and/or traffic

- Limits for a day, a week, a month and a total limit
- Operation in fixed day rate mode
- Operation in mixed modes
- Setting of rate per each hour at any day of the week and at holiday
- Setting of traffic cost (up to 24 traffic sizes).
- Setting of throughput for the subscriber.
- Per sec or per min charging
- Holidays
- Activation from the first connection and for the set period of time.
- Setting of login expiry date.
- Setting of login validity start time.
- Possibility to use the service on credit.
- Corporate bills (*)
- Login blocking
- Assignment of fixed IP addresses for each login
- Back call number assignment for each login
- Complete information about the connections (statistics)
- Easy and flexible system management via the web interface
- Operation with prepaid cards
- Limit for the access time and date
- "Live" statistics support (**)
- Subscriber disconnection (reset) at various conditions (**)
- Any methods of password checking (by means of other modules of the radius server)
- Individual subscriber price lists (version 2.0 and above)
- Different price lists for groups and subscribers for different access servers (version 2.0 and above).
- Concurrent prepaid and postpaid modes.
- Subscriber authorization by automatic number identifier.
- Subscriber authorization by IP address.
- Subscriber authorization by mac address.

* made possible due to other components.

** introduced for some hardware access servers

Service scopes

Billing is based on the scope of rendered services, which can be determined by the total duration of the subscriber sessions (time logging), subscriber traffic size (traffic logging), number of sessions and any combination thereof.

Logging of the number of sessions is charging of payment from the subscriber for each executed session. Payment is charged in accordance with the packet settings (individual subscriber settings) at session start.

Traffic logging is counting of the traffic size passed from/to the subscriber through the access server. It is determined by the price list of the service.

The following types of traffic can be logged: incoming traffic to the access server, outgoing traffic from the access server, total traffic (incoming + outgoing), maximum traffic (from the incoming and outgoing traffic), minimum traffic.

It should be noted that the maximum and minimum traffic is determined for the period of time between arrivals of alive packets from the access server and therefore it is inaccurate to speak of the maximum incoming or minimum outgoing traffic.

Service cost calculation by the traffic size can be performed by two methods as follows:

- Depending on the time of the day/day of the week (irrespective of the traffic size).
- Depending on the traffic size (irrespective of the time of the day/day of the week).

In both cases service cost calculation is based on the packet price list, individual subscriber price list or access server price list.

In the first instance a price list with individual cost settings for incoming and outgoing traffic for each hour of each day of the week (including individual settings for each hour of the holiday) is used.

In the second instance a price list with the settings of used traffic limits and traffic cost (specific for incoming traffic/specific for outgoing traffic) within these limits is used. Up to 24 traffic sizes can be set. Traffic sizes are specified as the lower and the upper limit. In both cases the price per 1 Mb of the traffic is specified.

Example:

0 — 300 : 1.5 : 1.5

300 — 1000 : 1.4 : 1.4

1000 — 0 : 1.2 : 1.2

In this example a price list with three traffic sizes is shown: from 0 Mb to 300 Mb with the cost of incoming and outgoing traffic of 1 ruble 50 kopecks, from 300 Mb to 1,000 Mb with the traffic cost within this range of 1 ruble 40 kopecks and from 1,000 Mb upwards with the traffic cost of 1 ruble 20 kopeks.

Time logging is performed even if the service cost calculation is not time dependent. Session start time, session end time and session duration are logged as minimum.

Service cost calculation is based on the price list (packet price list, individual subscriber price list or access server price list). The price list contains information about the cost of 1 hour of service for each hour of the day, day of the week and for a "special" day.

Session duration is counted per second or rounded up to minutes.

HOW FreeNIBS OPERATES?

Authorization

Authorization is a very easy process. The system receives login and password, checks the availability of the subscriber in the database, compares the password, checks the accounts, password expiry dates and decides whether to grant access or not.

Everything starts with the arrival of the packet Access-Request from the subscriber (pppd, Cisco, etc).

The packet text looks like as follows:

```
Service-Type = Framed-User
Framed-Protocol = PPP
User-Name = "test"
User-Password = "pass"
NAS-IP-Address = 192.168.0.1
NAS-Port = 5
```

Radius starts the process "authorize".

The billing system itself is interested only in the login and password and maybe also a little bit in the port and the access server IP.

Login and password are checked and if something is not correct Access-Reject does not grant access.

If all the parameters are correct the relevant attributes are set and sent as Access-Accept.

```
Service-Type = Framed-User
Framed-Protocol = PPP
Framed-IP-Address = 192.168.1.5
Framed-Compression = Van-Jacobson-TCP-IP
Idle-Timeout = 900
Session-Timeout = 3600
```

The billing system can add the following attributes in response:

- assigned IP (Framed-IP-Address);
- number for callback (Callback-Number);
- session timeout (Session-Timeout).

If the current time, price per time unit and the funds on account are known then it is possible to calculate the time, which corresponds to the funds on account, and set Session-Time for this time.

If subscriber does not terminate the session then the access server will disconnect him as soon as this time expires.

In case of the charge per traffic it is impossible to set an exact timeout. You can only try and work out some approximate values based on the analysis of the traffic size during the previous connection, calculate an average data transfer rate, etc.

You can also take some average data transfer rate, e. g. 21,000, and calculate the timeout for this rate.

However, these are only approximations and the subscribers can be disconnected either too early or too late.

There were some trial variants of the parameter Session-Octets-Limit for setting traffic limit per session. Nevertheless, not all the access servers support this feature.

The detailed description of the above procedure is as follows:

Module rlm_nibs is invoked. Function rlm_nibs_authorize is executed.

Subscriber availability in the database is checked and if the subscriber is found, then specific attributes are added (nibs_add_attrs function):

- Address assignment for the subscriber (Framed-IP-Address, if necessary packets.framed_ip or users.framed_ip).
- Assignment of the netmask for the user (Framed-IP-Netmask, if necessary packets.framed_ip or users.framed_ip).
- Addition of a callback number (Callback-Number, if necessary users.callback_number).
- Setting of maximum idle timeout (time, upon which expiry the session will be terminated if no packets arrive to/from the subscriber) for the session (Idle-Timeout, if necessary packet.idle_timeout).
- Setting of the number of simultaneous connections (Simultaneous-Use, if necessary packet.simultaneous_use).
- Setting of the maximum session timeout (Session-Timeout, if necessary packet.session_timeout or the calculated time).
- Addition of Port-Limit, if necessary (packet.port_limit).
- Addition of Huntgroup-Name, if necessary (packet.huntgroup_name).
- Addition of other attributes (packet.other_params).

Check for setting (Auth-Type == Nibs). If not, then Auth-Type will be set to Nibs (Auth-Type = Nibs). If Auth-Type is set to Nibs the module will return OK.

If the subscriber is not found in the database:

- If Auth-Type is set to Nibs (Auth-Type == Nibs), then the access is denied and the module will return REJECT.
- If Auth-Type is not equal to Nibs, then the module will return NOOP (continue checking with other modules).

If the process “authorize” returns OK, then Radius starts the process “authenticate”. If Auth-Type is set to Nibs, then the module rlm_nibs is invoked. Function rlm_nibs_authenticate is executed:

Subscriber availability check is performed. If the subscriber is found, then the password is checked. If the password does not correspond to the password in the database, then the access is denied and the module will return REJECT. Otherwise the module will return OK (subscriber access is allowed).

If the process “authenticate” returns OK, then the process “post-auth” will be started. Filter `nibs_filter` is started, which performs the following checks:

- If the subscriber uses the account (`packets.tos > 0`), then the filter checks whether there is any deposit on the account (`(users.deposit + users.credit) > 0`). If there is no deposit, then the access is denied (error 1).
- If the subscriber is not activated (`users.activated == 0`), then the filter checks whether the subscriber addition date is more than the current date (`users.add_date > now()`) – deny access (error 31)
- If the subscriber addition date is equal to zero (`users.add_date == 0`), then the addition date will be set to the current date (`users.add_date = now()`).
- If the activation time is equal to zero (`packet.activation_time == 0`), then the login expiry time is set to zero and ignored (`users.expired = 0`).
- If the activation time is more than zero (`packet.activation_time > 0`) and if the date for the activation is less than the login expiry date (`(now() + packets.activation_time) < users.expired`), then the expiry date is substituted with the new date (`users.expired = (now() + packets.activation_time)`).

Subscriber activation (`users.activated = 1`):

- If the login expiry date is not equal to zero and if it is less or equal to the current date (`users.expired <= now()`), then the access is denied (error 2).
- If the subscriber is blocked (`users.blocked = 1`), then the access is denied (error 3).
- If the total time limit is more than zero (`packets.total_time_limit > 0`) and if the total sum of the subscriber’s time is more or equal to the limit (`users.total_time >= packets.total_time_limit`), then the access is denied (error 11).
- If the month time limit is more than zero (`packets.month_time_limit > 0`), then the time for the current calendar month is calculated in accordance with the table “actions”. If the calculated time is more or equal to the limit (`%month_time% >= packets.month_time_limit`), then the access is denied (error 12).
- If the week time limit is more than zero (`packets.week_time_limit > 0`), then the time for the current week is calculated in accordance with the table “actions”. If the calculated time is more or equal to the limit (`%week_time% >= packets.week_time_limit`), then the access is denied (error 13).
- If the day time limit is more than zero (`packets.day_time_limit > 0`), then the time for the current day is calculated in accordance with the table “actions”. If the calculated time is more or equal to the limit (`%day_time% >= packets.day_time_limit`), then the access is denied (error 14).
- If the total traffic limit is more than zero (`packets.total_traffic_limit > 0`) and if the total subscriber traffic is more or equal to zero (`users.total_traffic >= packets.total_traffic_limit`), then the access is denied (error 21).
- If the month traffic limit is more than zero (`packets.month_traffic_limit > 0`), then the time for the current calendar month is calculated in accordance with the table “actions”. If the calculated traffic is more or equal to the limit (`%month_traffic% >= packets.month_traffic_limit`), then the access is denied (error 22).

- If the week traffic limit is more than zero (`packets.week_traffic_limit > 0`), then the time for the current week is calculated in accordance with the table "actions". If the calculated traffic is more or equal to the limit (`%week_traffic% >= packets.week_traffic_limit`), then the access is denied (error 23).
- If the day traffic limit is more than zero (`packets.day_traffic_limit > 0`), then the time for the current day is calculated in accordance with the table "actions". If the calculated traffic is more or equal to the limit (`%day_traffic% >= packets.day_traffic_limit`), then the access is denied (error 24).
- If the login time is limited (`packets.login_time[0] != 0`) and if the allowed login time is not equal to the current time (`packets.login_time != now()`), then the access is denied (error 40).

What is the format of the parameter login_time? If it is necessary to set the interval for the night tariff (e. g. from 23:00 to 7:00), will it be taken into account when calculating session-timeout, login_time?

*This parameter **will** be taken into account when calculating session-timeout.*

Parameter format is very easy: intervals divided by commas and consisting of the day and time (DDHHMM-HHMM) or just the day (DD).

The following days are supported:

Su - Sunday

Mo - Monday

Tu - Tuesday

We - Wednesday

Th - Thursday

Fr - Friday

Sa - Saturday

Wk – working days (= Mo, Tu, We, Th, Fr)

Al – all days

HI - holidays

Example:

Wk0000-0800,Sa,Su,HI2200-1000

It means access allowed from Monday till Friday only from 0 to 8 a.m., on Saturday and Sunday round-the-clock and on holidays from 10 p.m. to 10 a.m.

- Checking for the forbidden telephone number. If the number is forbidden then (error 50). In this respect the following section is to be reviewed in detail.

Description of black/white list system operation

Comparison of the telephone number received from the subscriber's equipment with the number stored in the database is performed from left to right, i. e. the numbers received from the public exchange are to be equal in length.

Symbol '?' in the mask coincides with any symbol of the telephone number whereas symbol '*' coincides with all the remaining symbols of the telephone number up to the end.

For instance, '8815245*' coincides with all the telephone numbers of the city of Murmansk, which begin with '45' (it is implied that the public exchange sends the numbers with the long distance prefix '8'. This can be different depending on the settings of your hardware and the public exchange of your telephone access carrier);

'8815245432?' coincides with the telephone numbers of the city of Murmansk, which begin with '45432' and end with any number.

*PLEASE NOTE that '88152*45' is actually equivalent to '88152*' because '*' coincides with all the remaining symbols of the telephone number up to the end (see above).*

Checking of group/subscriber binding is performed in accordance with the following rule:

- If uid==NULL and gid==NULL, then the rule refers to ALL subscribers; will be checked in the first place.
- If uid==NULL and gid==<integer>, then the rule refers to the subscribers of the group with gid=<integer>; will be checked in the second place, has higher priority as the first check and can change the result of the first check.
- If uid=<integer> and gid==NULL, then the rule refers to the subscriber with uid=<integer>; will be checked in the last place, has the highest priority and can change the result of the first two checks.
- If uid=<integer> and gid=<integer>, then the action is NOT DEFINED.

In case the latter two checks reveal the coincidence with the group or the subscriber and the 'permit' is equal to 'y', then by default the access will be FORBIDDEN, i. e. if there is at least one allowed number for the group or the subscriber, the access will be granted only from that number, if there are no other rules.

Checks are performed until the first coincidence in each of the three cases.

Example:

Uid	Gid	Phone	Permit
500	NULL	8152451000	Y
NULL	NULL	8152450000	N
100	NULL	8152541234	Y
NULL	1000	815254*	N

1. The software product will allow the subscriber with uid=500 to establish connection only from the number 8152451000.
2. The software product will prohibit any subscribers to establish connection from the number 8152450000.

In case there is a subscriber with uid=100 and gid=1000:

3. The software product will allow this subscriber to establish connection from the number 8152541234, although the access for his group from this number is forbidden.
- If there is an error during the check, then (error -1).
 - If everything is OK, then (error 0).

If the filter does not return 0, then the access is forbidden and the module will return REJECT. Otherwise the module will return OK.

If the process “post-auth” also returns OK, then the software product will allow the.

Accounting

ACCT START packet arrives. Radius starts the process “accounting”.

Module rlm_nibs is invoked. Function rlm_nibs_accounting is executed.

Subscriber availability is checked:

- If the subscriber is found and if statistics in the packet is disabled (packet.no_acct == 1), then the module will return OK.
- Session start entry is made in the table “actions”. The module returns OK.
- If there is no such subscriber in the database, then the module will return NOOP.

ACCT ALIVE packet arrives. Radius starts the process “accounting”.

Module rlm_nibs is invoked. Function rlm_nibs_accounting is executed.

Subscriber availability is checked:

- If the subscriber is found and if statistics in the packet is disabled (packet.no_acct == 1), then the module will return OK.
- If there is no session start entry in the table “actions” for this ALIVE packet, then such entry is added to this table.
- Function nibs_billing is executed: statistics and money counting. If the server does not return Acct-Terminate-Cause, then:
- Acct-Terminate-Cause is set to User-Request.
- Fields deposit, total_traffic and total_time are corrected in the table [Subscribers] (if necessary).
- Fields in_bytes, out_bytes, time_on and framed_ip are corrected in the table “actions”.
- If (%kill_by_alive% == 1), then the function nibs_filter is executed (ist operation is similar to the procedure described in the AUTH part).
- If (error != 0), then the function nibs_user_kill is started, which will invoke %kill_program%.
- The module will return OK.
- If there is no such subscriber in the database, then the module will return NOOP.

ACCT STOP packet arrives. Radius starts the process “accounting”.

Module rlm_nibs is invoked. Function rlm_nibs_accounting is executed.

Subscriber availability is checked and if the subscriber is found, then:

- If the subscriber is found and if statistics in the packet is disabled (packet.no_acct == 1), then the module will return OK.
- If there is no session start entry in the table “actions” for this STOP packet, then such entry is added to this table.

Function nibs_billing is executed: statistics and money counting.

- If the access server does not return Acct-Terminate-Cause, then Acct-Terminate-Cause is set to User-Request.
- Fields deposit, total_traffic and total_time are corrected in the table [Subscribers] (if necessary).
- Session start entry in the table “actions” is substituted with session end entry. The module will return OK.
- If there is no such subscriber in the database, then the module will return NOOP.

Web interface (control interface)

FreeNIBS/STArad control interface is a software component, which extends NETSHe control shell and is integrated therein.

This component operates in the standard browser with enabled JavaScript support. No additional software is required.

General NETSHe system concept

NETSHe system concept can be generally described as hierarchical relationships:

- System owner
- One or more service providers
- Several resellers for each provider.

Providers and resellers have authorized operators. Rights of individual operators in the system are adjustable.

Providers render one or more services (e. g. Internet access, hosting, equipment rental).

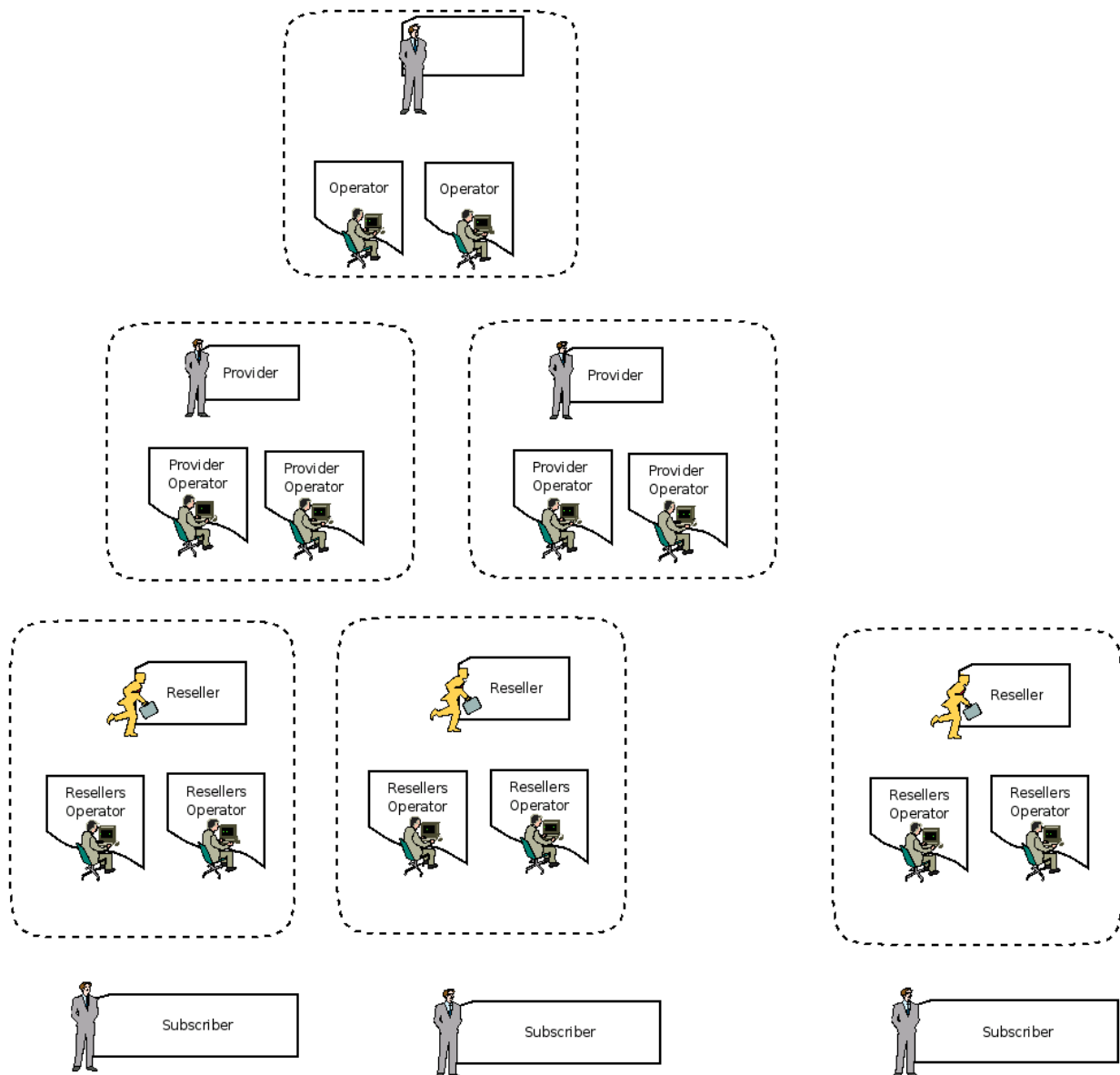
The rules and parameters of the providers' services are determined by the tariffs. The tariffs describe the payment procedure for the rendered services, payment dates and other service related parameters.

As a rule, resellers re-sell the services of the providers. Not all of the provider's tariffs can be accessible to them (and subsequently to the subscribers) and their operations in the system can be restricted.

The system has functions related to charging for the use of services and execution of settlements between the system owners and providers, providers and resellers, providers and subscribers.

There are several web interface levels in the system (see the diagram below):

- Access for system owners (root).
- Access for providers
- Access for reseller
- Restricted access for operator
- Access for subscriber (end user).



Getting started

In order to get started it is necessary to at least specify the required number of providers, operators, access servers (specific for each provider), tariffs (specific for each provider), specify days off (specific for each provider) and enter the required number of subscribers (specific for each provider).

Immediately after the system installation only root access is possible with login "admin" and password "admin".

Entry of companies, people, providers and other basic information

Remember that the system owner, any provider or reseller are companies (they are to be entered in the system as companies) and any operator is to be entered as "people".

In order to enter companies it is necessary to choose the menu item "Data->Companies".

The screenshot shows a web browser window titled "[NETSHe-0.1] - Opera". The address bar displays "http://localhost:5556/suite/root/index.php?action=sys.Companies.table_view&page=0". The page content includes a navigation bar with "New", "Home", "Help", and "Logout" links. Below this is a section titled "[Companies]" containing a table with 11 columns: #, Title, Organization, Tax num, Town, Street, Ext. add, Work phone, Fax, Cell phone, and Option. The table lists 12 companies, each with a unique ID and various details. A "Help" dialog box is overlaid on the table, showing icons for "New", "Edit", "Find", "Delete", and "Lock".

#	Title	Organization	Tax num	Town	Street	Ext. add	Work phone	Fax	Cell phone	Option
28	TEST_FOR	A company for tests	121212	Palma de Mallorca						
11	EKOPLC	EKOPLC	1	Palma-de-Mallorca						
20		HLG								
38	Altea Hills	Hotel Altea Hills	1	Palma						
26	Hoteles Costa Blanca	Hoteles Costa Blanca	1	Spain						
37	Villa Gadea	Hotel Villa Gadea								
14	Ifach	Ifach hotel								
13	Ingles	Ingles hotel								
17		INSOTEL								
4	OPTIMIZA	OPTIMIZA	1	Valencia						
15	Palace	Palace hotel	1	Palma						
24		TRASMEDITERRANEA								

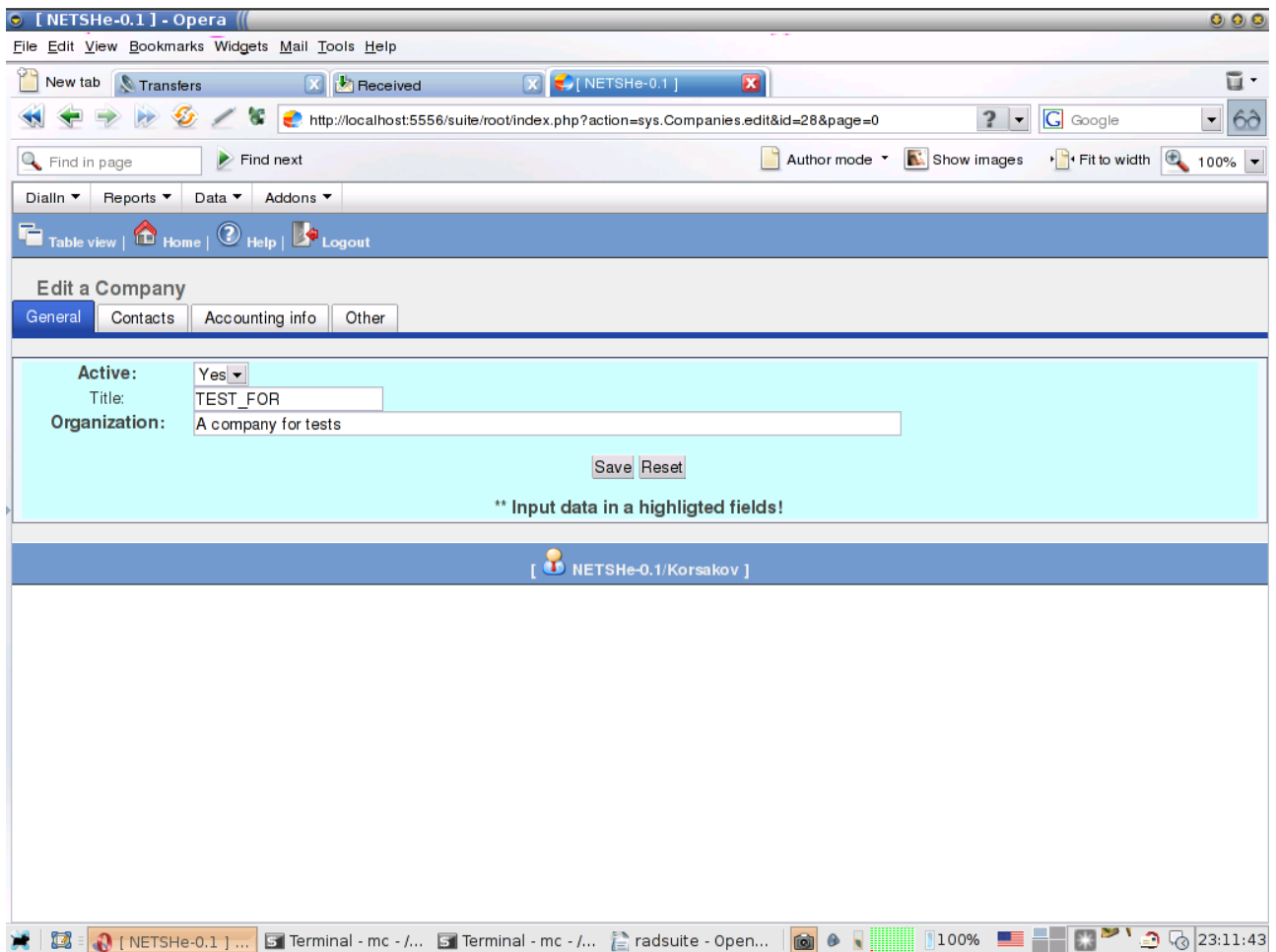
Total: 12

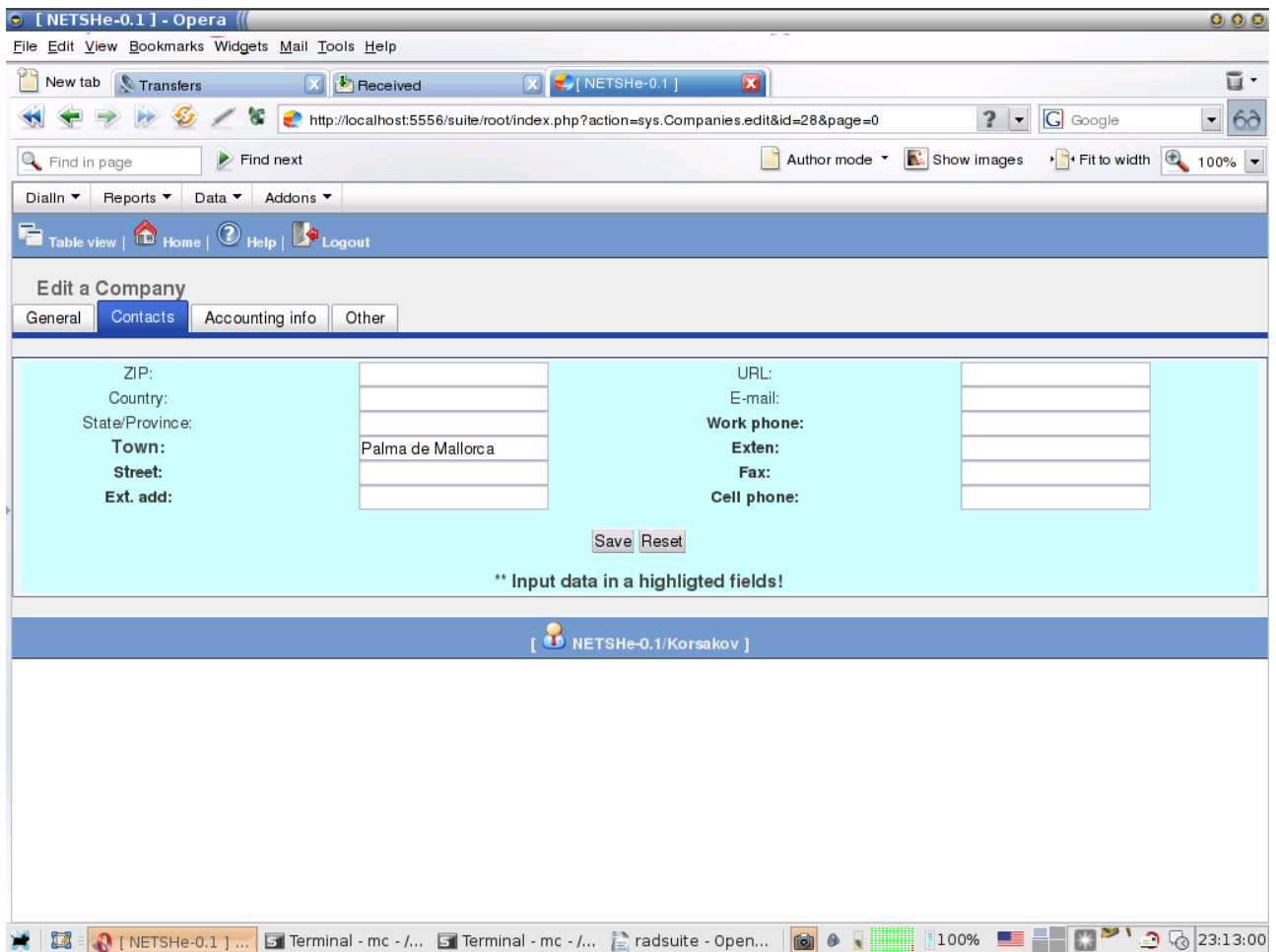
This will display all the companies in the system.

Here we can enter new companies, edit and delete existing companies.

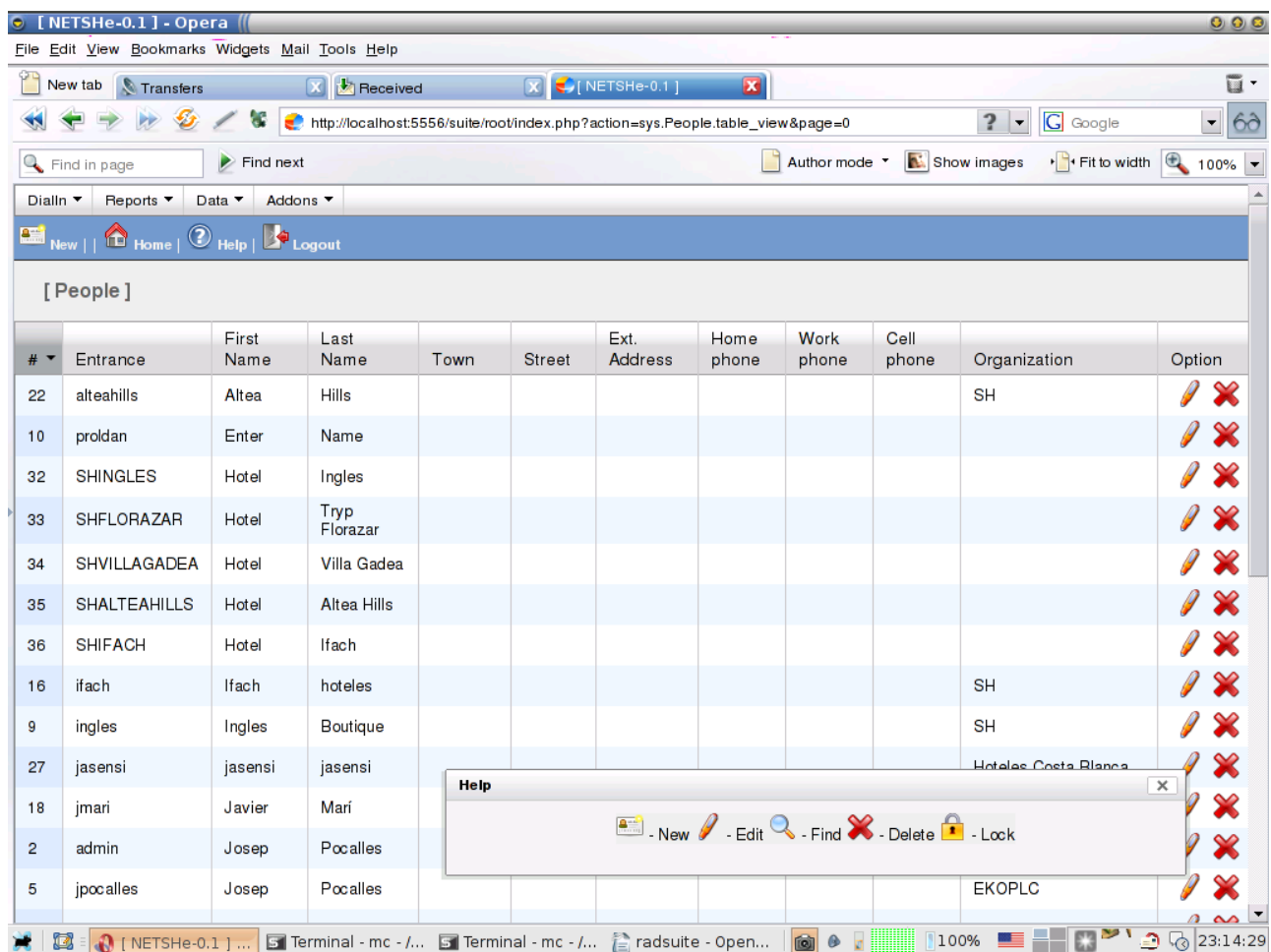
A typical window for entering a new company/editing the existing company has the following outlook:

Input of contact information of the company:





In order to enter people choose the menu item “Data->People”.



Procedure similar to the entry/deletion/editing of the companies.

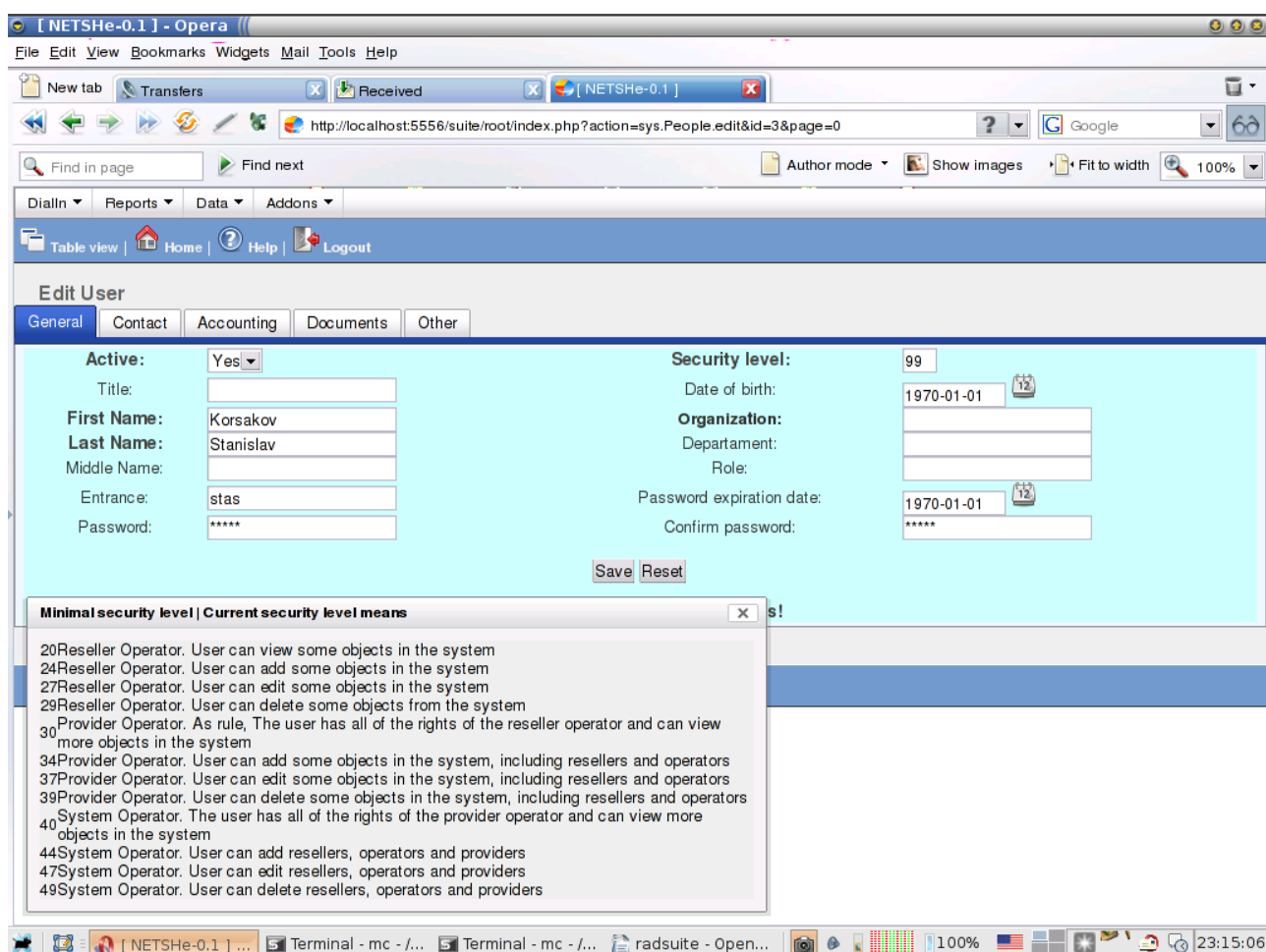
People input dialog has at least three important elements:

- login for entrance into the system
- password
- user privilege level (security level).

Security level set the following features:

Numerical value	User role in the system (who the user can act as)
20	Operator. User can view some objects in the system
24	Operator. User can add some objects in the system
27	Operator. User can edit some objects in the system
29	Operator. User can delete some objects from the system
30	Provider operator. As a rule, the user has all the rights of the operator

Numerical value	User role in the system (who the user can act as)
34	Provider operator. User can add some objects in the system, including operators and resellers
37	Provider operator. User can edit some objects in the system, including operators and resellers
39	Provider operator. User can delete some objects from the system, including operators and resellers
40	System operator. As a rule, the user has all the rights of the provider operator
44	System operator. User can add some objects in the system, including operators and resellers
47	System operator. User can edit some objects in the system, including operators and resellers
49	System operator. User can delete some objects from the system, including operators and resellers

























































Security level “99” allows to perform any operations in the system.

Then at least one provider is to be entered.

Opera browser window showing the [NETSHe-0.1] application. The address bar displays `http://localhost:5556/suite/root/index.php?action=sys.Providers.table_view&page=0`.

The application interface includes a menu bar (File, Edit, View, Bookmarks, Widgets, Mail, Tools, Help) and a toolbar with buttons for New tab, Transfers, Received, and [NETSHe-0.1].










The main content area displays a table titled [Providers] with the following data:

#	Title	Company	CEO	CFO	CAO	CTO	Responsible person	Document	Option
3	OPTIMIZA	OPTIMIZA	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles		        
8	EKOPLC	EKOPLC	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles		        
10	INSOTEL	INSOTEL	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles		        
11	HLG	HLG	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles		        
12	TRASMEDITERRANEA	TRASMEDITERRANEA	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles		        
13	Hoteles Costa Blanca	Hoteles Costa Blanca	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles		        

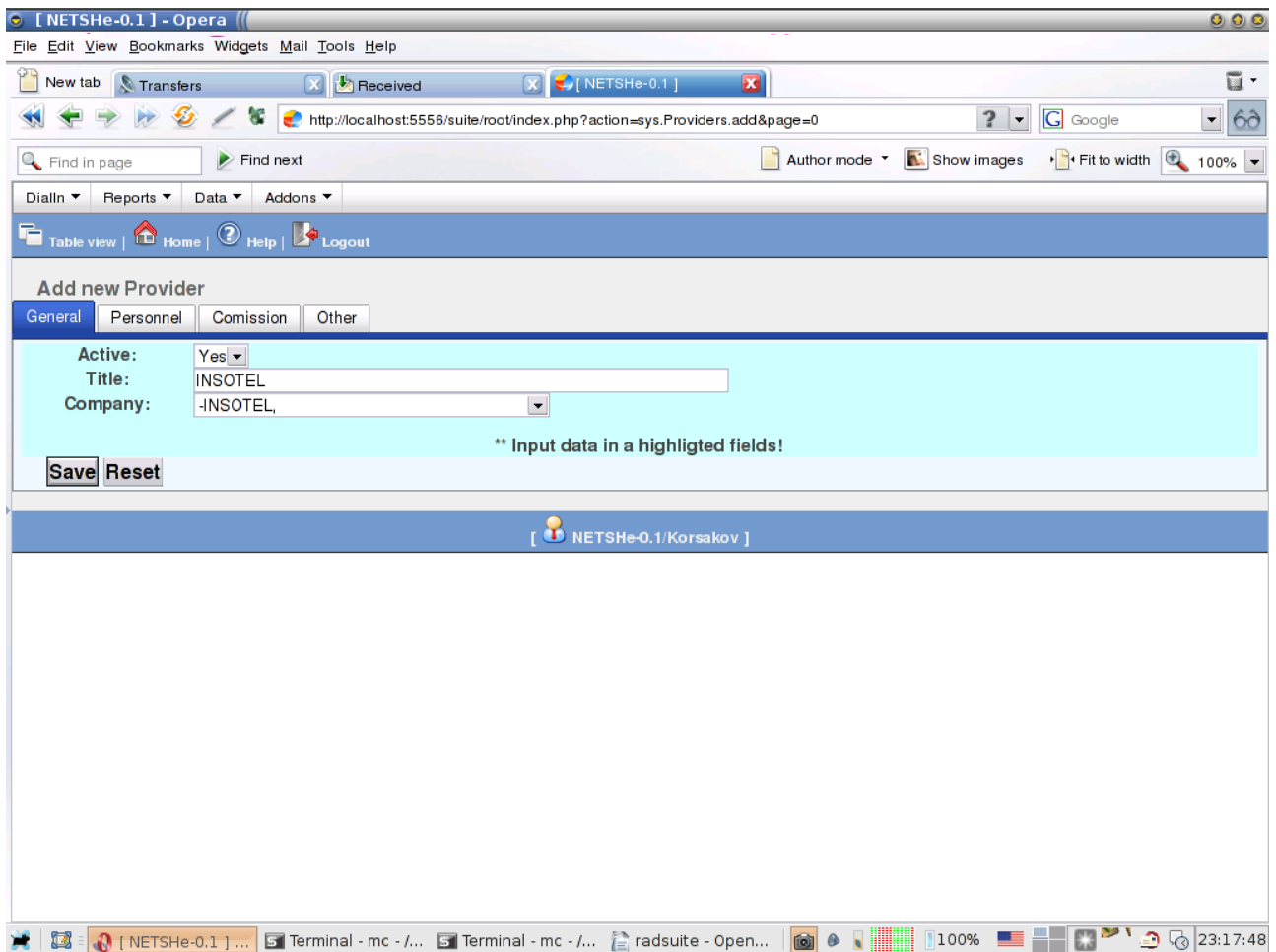
Total: 6

The application footer shows [NETSHe-0.1/Korsakov].

A Help dialog box is open, displaying the following icons and actions:

-  - New
-  - Edit
-  - Find
-  - Delete
-  - Lock
-  - Operators
-  - Resellers
-  - Services
-  - Equipment

The taskbar at the bottom shows the Opera browser window, Terminal - mc - /..., and radsuite - Open... with a system clock of 23:16:02.



And the operator for this provider.

File Edit View Bookmarks Widgets Mail Tools Help

New tab Transfers x Received x [NETSHe-0.1] x

http://localhost:5556/suite/root/index.php?action=sys.Operators.table_view&id=3&page=0

Find in page Find next Author mode Show images Fit to width 100%

DialIn Reports Data Addons

New Home Help Logout

[Operators for (3)]

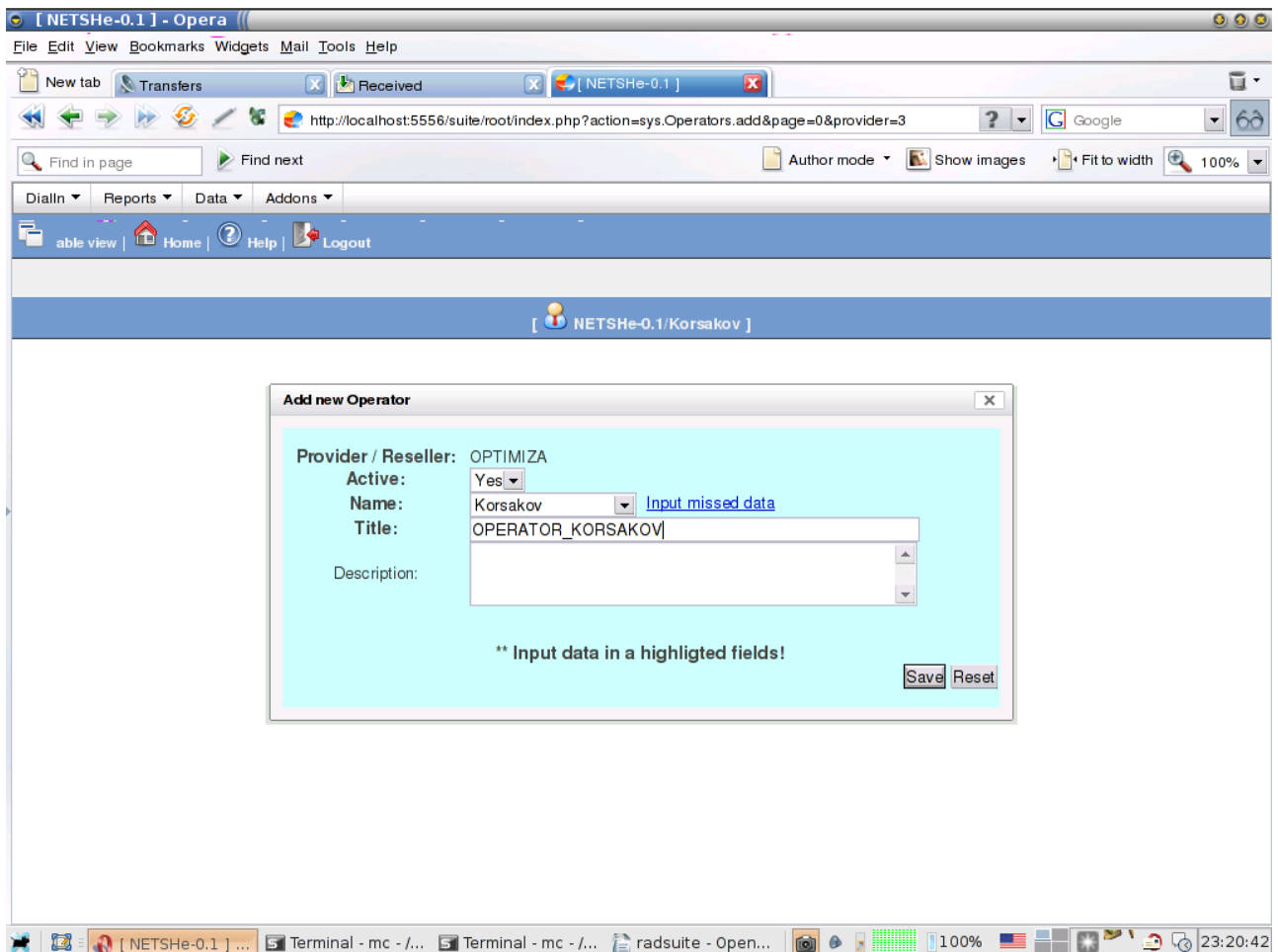
#	Title	Name	Provider / Reseller	Responsible person	Option
10	josevi	Josevi Enguidanos	OPTIMIZA	Josep Pocalles	
3	Director Comercial	Raul Hurtado	OPTIMIZA	Josep Pocalles	
7	tryp_florazar	Tryp Florazar	OPTIMIZA	Josep Pocalles	
27	Informatica	Please, enter name Please, enter surname	OPTIMIZA	Korsakov Stanislav	

Total: 4

Help

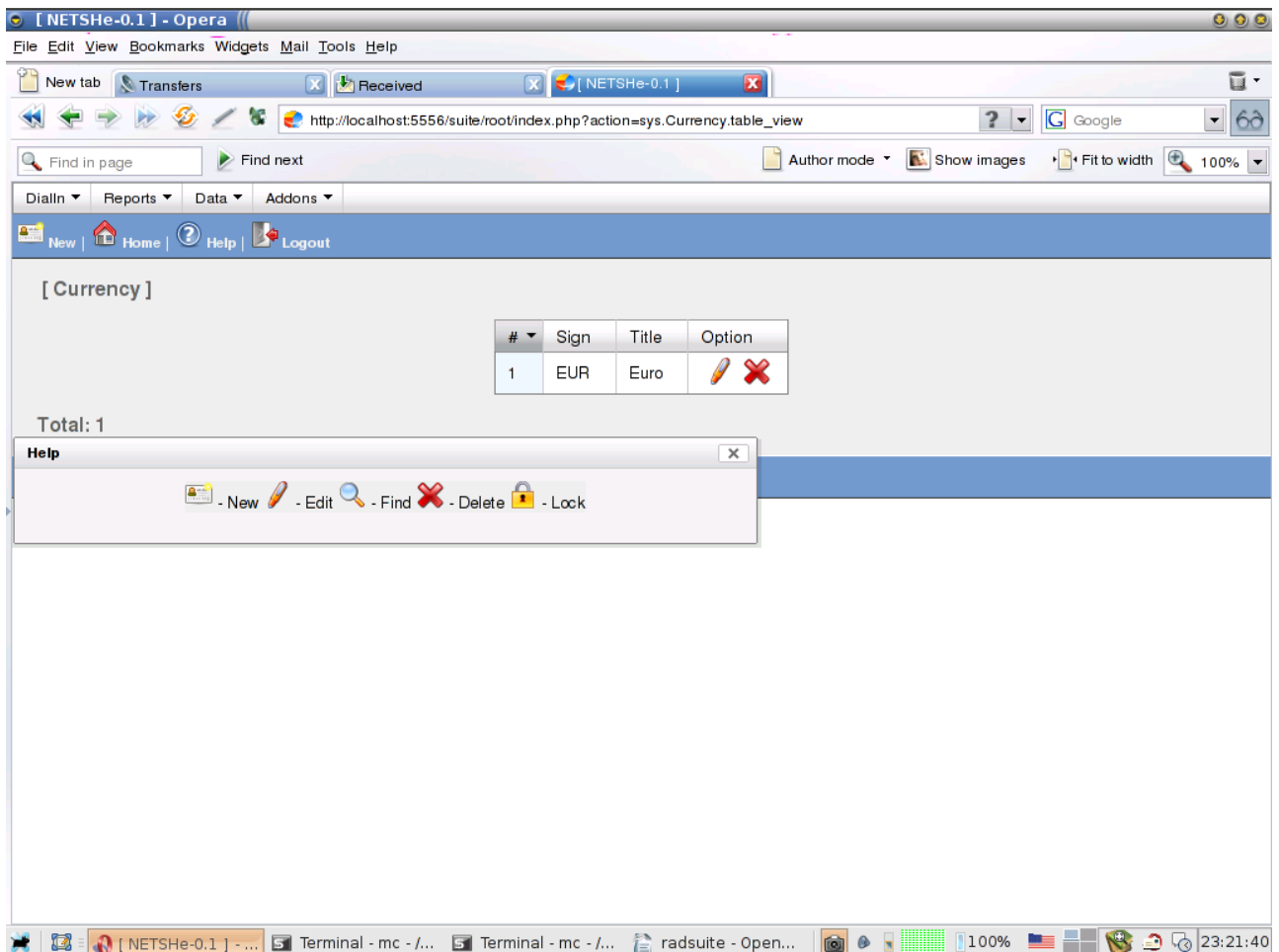
- New - Edit - Find - Delete - Lock

Terminal - mc - /... Terminal - mc - /... radsuite - Open... 100% 23:19:24

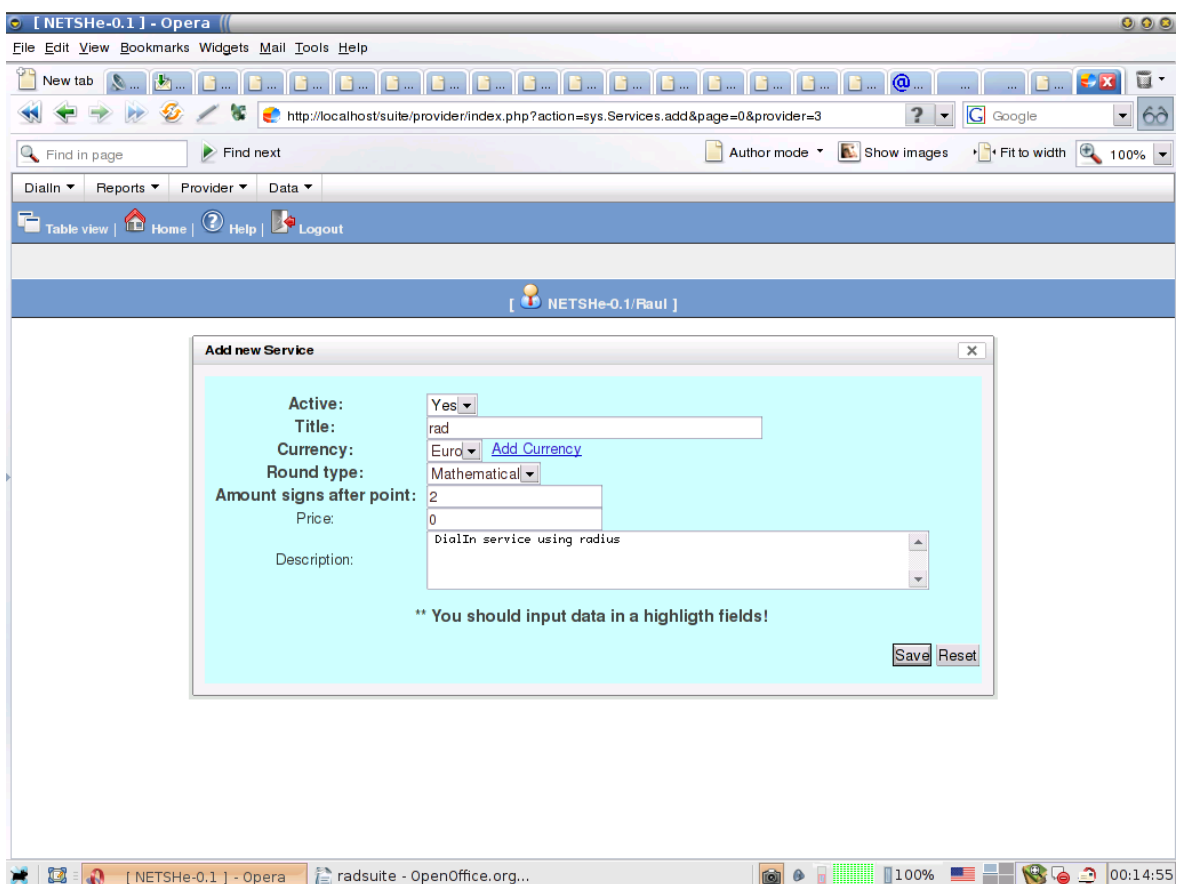


Remember that the entry of resellers and operators is to be performed from the provider's interface.

After the entry of providers and operators it is necessary to specify the currencies and list of services for each provider from the provider's interface.

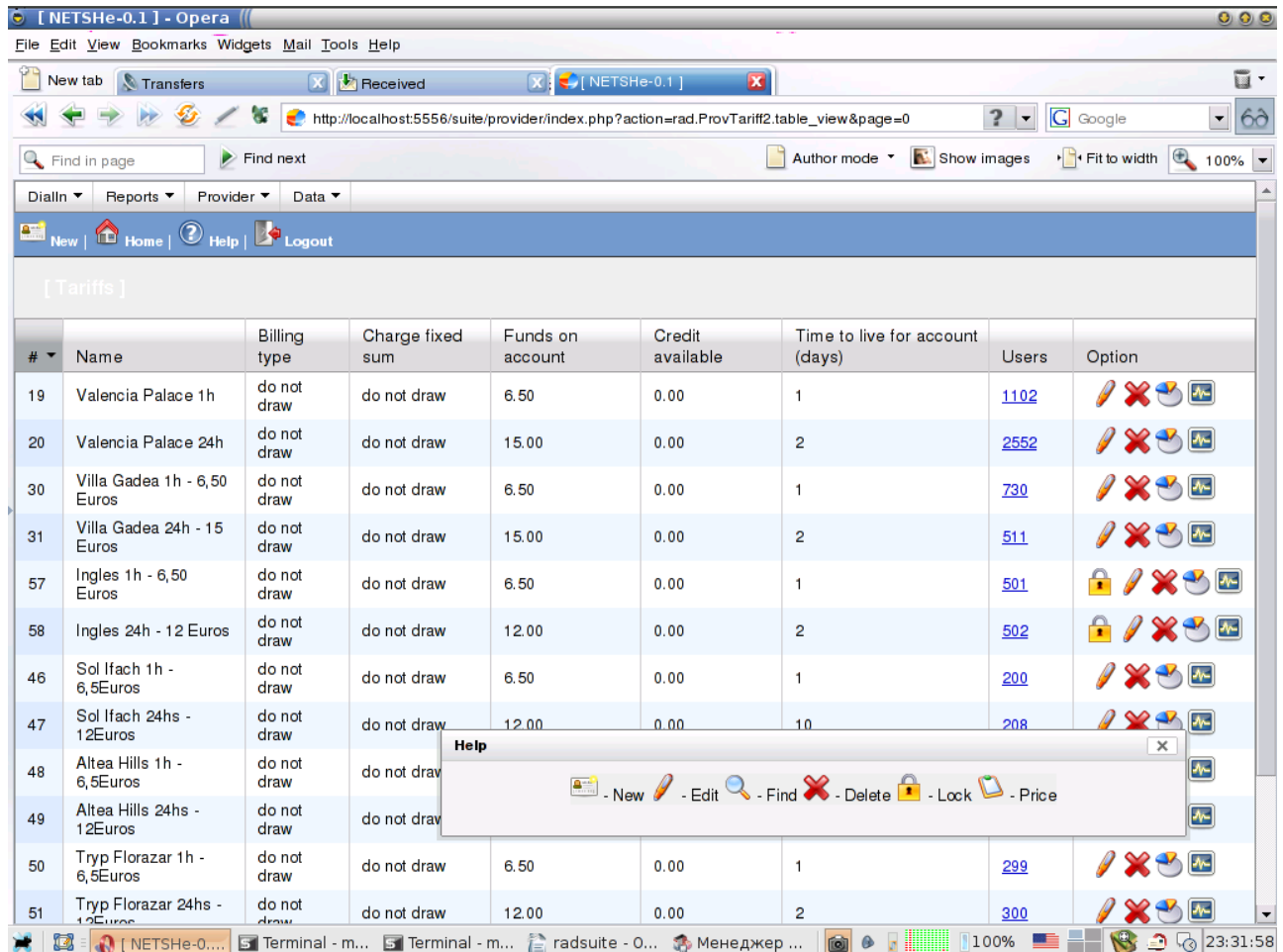


Definition of services.











































































It should be noted that resellers and their operators can be entered at any time as required.

Tariffs for the subscribers



[Tariffs]

#	Name	Billing type	Charge fixed sum	Funds on account	Credit available	Time to live for account (days)	Users	Option
19	Valencia Palace 1h	do not draw	do not draw	6.50	0.00	1	1102	     
20	Valencia Palace 24h	do not draw	do not draw	15.00	0.00	2	2552	     
30	Villa Gadea 1h - 6,50 Euros	do not draw	do not draw	6.50	0.00	1	730	     
31	Villa Gadea 24h - 15 Euros	do not draw	do not draw	15.00	0.00	2	511	     
57	Ingles 1h - 6,50 Euros	do not draw	do not draw	6.50	0.00	1	501	     
58	Ingles 24h - 12 Euros	do not draw	do not draw	12.00	0.00	2	502	     
46	Sol Ifach 1h - 6,5Euros	do not draw	do not draw	6.50	0.00	1	200	     
47	Sol Ifach 24hs - 12Euros	do not draw	do not draw	12.00	0.00	10	208	     
48	Altea Hills 1h - 6,5Euros	do not draw	do not draw	do not draw	0.00			     
49	Altea Hills 24hs - 12Euros	do not draw	do not draw	do not draw	0.00			     
50	Tryp Florazar 1h - 6,5Euros	do not draw	do not draw	6.50	0.00	1	299	     
51	Tryp Florazar 24hs - 12Euros	do not draw	do not draw	12.00	0.00	2	300	     

List of tariffs in the system [Tariffs]. The following operations are available: creation of a new tariff, editing of an existing tariff, deletion of the tariff and statistics.

The table contains basic information about the tariffs: tariff name, billing type (time, traffic, etc.), sum on the account of the newly created subscriber of the tariff, number of subscribers for each tariff, etc.

It should be noted that adding of a new tariff is to be performed ONLY from the provider's or reseller's interface with relevant rights.

Note 1: Tariff properties set to "0" or empty fields mean that there are no restrictions with regards to the given parameter and (or) the access server settings are used.

Note 2: Setting of the same subscriber attributes to the values, which differ from the values of the tariff results in the use of the values of the subscriber attributes.

Example: If there is no individual price list defined for the subscriber, then the price list of the tariff is used.

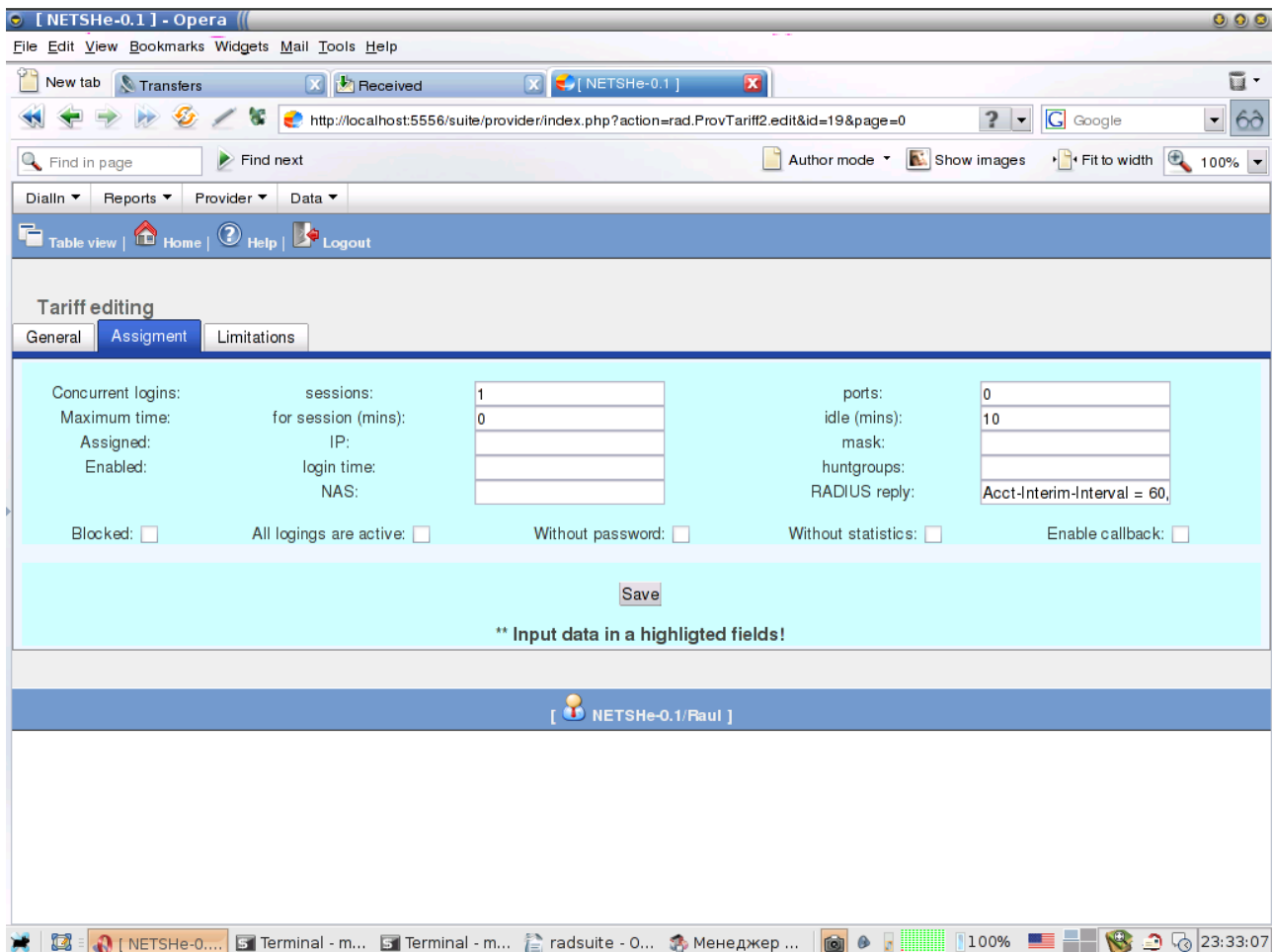
Tariff editing form has three tabs: General, Assignments, Limitations.

The screenshot shows a web browser window titled "[NETSHe-0.1] - Opera". The address bar displays the URL: `http://localhost:5556/suite/provider/index.php?action=rad.ProvTariff2.edit&id=19&page=0`. The browser's menu bar includes File, Edit, View, Bookmarks, Widgets, Mail, Tools, and Help. The toolbar shows navigation buttons, a search bar, and a zoom level of 100%. The page content is titled "Tariff editing" and features three tabs: "General", "Assignment", and "Limitations". The "General" tab is active, displaying a form with the following fields:

Name:	Valencia Palace 1h
Billing type:	do not draw
traffic:	do not count
money from account:	draw and check funds
Charge fixed sum:	do not draw
Fixed sum:	0.00
Funds on account:	6.50
Credit available:	0.00
Time to live for account (days):	1

Below the form is a "Save" button and a message: "** Input data in a highlighted fields!". The footer of the page shows the user's name and the application version: "[NETSHe-0.1/Paul]". The taskbar at the bottom of the screen shows several open applications, including "Terminal - m...", "radsuite - O...", and "Менеджер ...".

The "General" tab in the above screenshot provides the following information: name of the tariff, billing type (billing is not performed), traffic type (traffic is not counted), fixed sum (it is not charged), etc. All newly added subscribers of this tariff will have 2 units of money on their account (rubles, Euros, dollars, etc.). Credit is not available for the subscribers of this tariff. Login effective period for this tariff is 1 day from the start of the first session.



“Assignments” tab contains information with regards to the service rules, such as (see example above):

- Number of concurrent logins (sessions): 1;
- There are no limitations with regards to the number of concurrently used access server ports;
- Maximum session time is not explicitly limited;
- Idle time is limited to 10 minutes;
- Access server settings are used to assign the address to the subscriber (empty field means that RADIUS does not participate in the process of address assignment).
- There are no limitations with regards to connection time and access server;
- Tariff is not blocked;
- Subscribers are not enabled by default;
- In order to gain access the subscriber is to enter password;
- Statistics is tracked for every subscriber of the tariff;
- Callback is disabled.

NETSHe-0.1 - Opera

File Edit View Bookmarks Widgets Mail Tools Help

http://localhost:5556/suite/provider/index.php?action=rad.ProvTariff2.edit&id=19&page=0

Find in page Find next Author mode Show images Fit to width 100%

DialIn Reports Provider Data

Table view Home Help Logout

Tariff editing

General Assignment **Limitations**

Limits	per Day	per Week	per Month	Total
Time (mins):	0	0	0	60
Traffic (MB):	0	0	0	0
Money:	0.00	0.00	0.00	0.00

Save

**** Input data in a highlighted fields!**

[NETSHe-0.1/Paul]

“Limitations” tab contains information about the subscriber activity within the lifetime of the login and particularly the following (see the example above):

- Limit for the accumulated duration of the sessions per day, per week, per month and the total limit;
- Limit for subscriber traffic in megabytes (per day, per week, per month and total). Please remember that in this case the billing type is to be traffic (incoming, outgoing, maximum, minimum, incoming+outgoing or total);
- Limit for spent money (per day, per week, per month and the total limit).

Any limits in any combination can be set.

The following rule is to be taken into account when setting limits:

Day limit \leq week limit \leq month limit \leq total limit

Because billing type is not set for this tariff, it lacks any price list.

How the subscriber of this tariff is managed?

Subscriber (login) deactivation time is set at the start of the first session. This time is set as current time + 24 hours.

Session timeout is calculated *function min* (subscriber deactivation time — current time, 20 minutes).

Session start and end time is stored (updated) for each subscriber session.

Now we can review the tariff with billing type “time”.

The screenshot shows the 'Tariff editing' interface in the NETSHe-0.1 application. The 'General' tab is selected, and the form contains the following fields and values:

Field	Value
Name:	7 days
Billing type:	time
traffic:	do not count
money from account:	draw and check funds
Charge fixed sum:	do not draw
Fixed sum:	0.00
Funds on account:	30.00
Credit available:	0.00
Time to live for account (days):	7

Below the form is a 'Save' button and a message: "** Input data in a highlighted fields!". The bottom of the interface shows the user's name 'NETSHe-0.1/Josep'.

Billing type “time” is specified on the tab “General”. Time to live for account is set to 7 days. When a new subscriber is created the sum on account will be 30 units of money. “Assignments” tab has the same settings as the above shown tariff.

FileEditViewBookmarksWidgetsMailToolsHelp

New tabTransfersReceived[NETSHe-0.1]

http://localhost:5556/suite/provider/index.php?action=rad.ProvTariff2.edit&id=10&page=0

Google

Find in pageFind nextAuthor modeShow imagesFit to width100%

DialInReportsProviderData

Table viewHomeHelpLogout

Tariff editing

GeneralAssignmentLimitations

Limits	per Day	per Week	per Month	Total
Time (mins):	0	0	0	10080
Traffic (MB):	0	0	0	0
Money:	0.00	0.00	0.00	0.00

Save

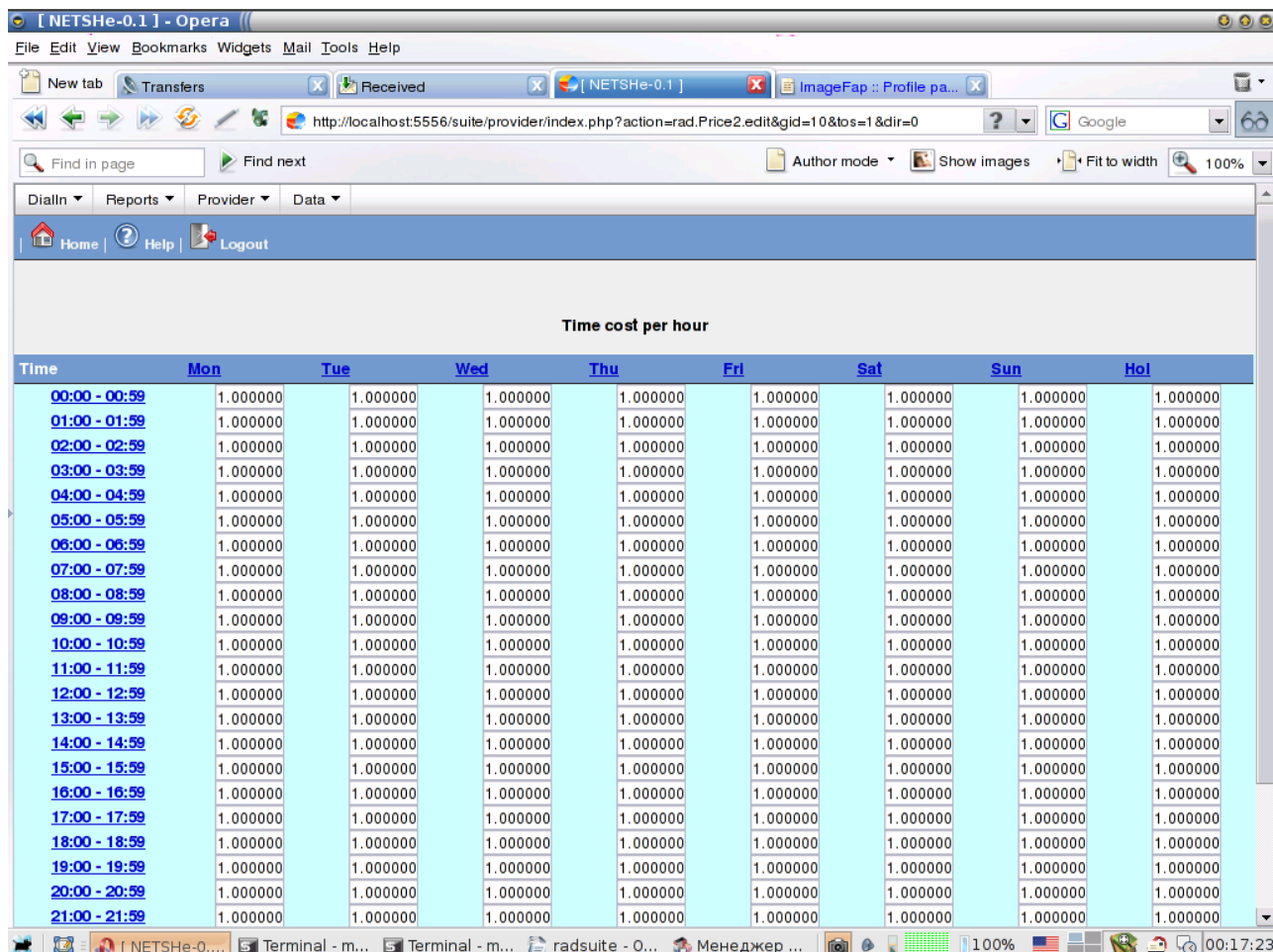
** Input data in a highlighted fields!

[NETSHe-0.1/Josep]

NETSHe-0.1...Terminal - m...Terminal - m...radsuite - 0...Менеджер ...100%00:11:09

Total limit in the tab “Limitations” is set to 10080 minutes.

The main distinction of such tariff is the price list:



The screenshot shows a web browser window with the address bar displaying `http://localhost:5556/suite/provider/index.php?action=rad.Price2.edit&gid=10&tos=1&dir=0`. The page title is "[NETSHe-0.1]". The browser's address bar shows the URL. The page content includes a navigation bar with links for Home, Help, and Logout. Below the navigation bar is a table titled "Time cost per hour". The table has columns for Time, Mon, Tue, Wed, Thu, Fri, Sat, Sun, and Hol. Each cell in the table contains the value 1.000000. The table is displayed in a light blue color scheme.

Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Hol
00:00 - 00:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
01:00 - 01:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
02:00 - 02:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
03:00 - 03:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
04:00 - 04:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
05:00 - 05:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
06:00 - 06:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
07:00 - 07:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
08:00 - 08:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
09:00 - 09:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
10:00 - 10:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
11:00 - 11:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
12:00 - 12:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
13:00 - 13:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
14:00 - 14:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
15:00 - 15:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
16:00 - 16:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
17:00 - 17:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
18:00 - 18:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
19:00 - 19:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
20:00 - 20:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
21:00 - 21:59	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Price list is a table where every hour of the day in every day of the week is assigned a certain price. This price is specified in units of money per hour.

Note: Mouse click on the column leads to automatic filling of all the cells in that column with the value taken from the top cell.

Note: Mouse click on the row leads to automatic filling of all the cells in that row with the value taken from the left cell.

Billing procedure for a subscriber to this tariff

“Time to live” for the subscriber is set to 30 days from the moment of the first session start.

Sum on the account of the subscriber is reduced in accordance with the duration of the sessions and the price list for the connection time.

The service is aborted as soon as one of the following conditions arises:

- 30 days from the moment of the first session start expired;
- Total duration of all the sessions amounts to or exceeds 10080 minutes
- Subscriber’s balance (sum on the account) is zero or negative.

Let us now review the tariff with billing type “traffic”.

The screenshot shows a web browser window titled "[NETSHe-0.1] - Opera". The address bar displays the URL: `http://localhost:5556/suite/provider/index.php?action=rad.ProvTariff2.edit&id=63&page=0`. The browser's menu bar includes File, Edit, View, Bookmarks, Widgets, Mail, Tools, and Help. The toolbar shows navigation buttons, a search bar, and a Google search button. The page content is divided into a header with navigation links (DialIn, Reports, Provider, Data) and a main section titled "Tariff editing".

The "Tariff editing" section has three tabs: "General", "Assignment", and "Limitations". The "General" tab is active, showing a form with the following fields:

Name:	Test traffic tariff 2
Billing type:	traffic
traffic:	incoming
money from account:	draw and check funds
Charge fixed sum:	do not draw
Fixed sum:	0.00
Funds on account:	10
Credit available:	1
Time to live for account (days):	0

Below the form is a "Save" button and a message: "** Input data in a highlighted fields!". The footer of the page shows the user name "[NETSHe-0.1/Josep]".

Billing type “traffic” is specified on the tab “General”. It is also specified that the incoming traffic is to be counted. When a new subscriber is created the sum on account will be 10 units of money. The subscriber’s account can be overdrawn in the amount of 1 unit of money.

FileEditViewBookmarksWidgetsMailToolsHelp

New tabTransfersReceived[NETSHe-0.1]

http://localhost:5556/suite/provider/index.php?action=rad.ProvTariff2.edit&id=63&page=0

Google

Find in pageFind nextAuthor modeShow imagesFit to width100%

DialInReportsProviderData

Table viewHomeHelpLogout

Tariff editing

GeneralAssignmentLimitations

Limits	per Day	per Week	per Month	Total
Time (mins):	0	0	0	0
Traffic (MB):	0	0	0	0
Money:	0.00	0.00	0.00	0.00

Save

** Input data in a highlighted fields!

[NETSHe-0.1/Josep]

NETSHe-0.1Terminal - mc - ./...Terminal - mc - ./...radsuite - Open...100%00:20:30

There are no additional limitations for the subscriber to this tariff.

Incoming traffic costs per MB

Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Hol
00:00 - 00:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
01:00 - 01:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
02:00 - 02:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
03:00 - 03:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
04:00 - 04:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
05:00 - 05:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
06:00 - 06:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
07:00 - 07:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
08:00 - 08:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
09:00 - 09:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
10:00 - 10:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
11:00 - 11:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
12:00 - 12:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
13:00 - 13:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
14:00 - 14:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
15:00 - 15:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
16:00 - 16:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
17:00 - 17:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
18:00 - 18:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
19:00 - 19:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
20:00 - 20:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000
21:00 - 21:59	2.000000	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000

Price list for this tariff contains prices for 1 Mb of the traffic. Prices are specified for every hour of the day and for every day of the week including days off.

In case of incoming and outgoing traffic counting as well as traffic + time counting the price list editing form automatically invokes the corresponding parts. Below is the description of the subscriber billing procedure in case of such tariff.

Service value is calculated continuously during the session (upon arrival of START, ALIVE and STOP packets). The calculated amount is deducted from the run on the account.

As soon as the sum on the account is zero or less {0-sum_credit} the software application will try to disconnect the subscriber from the service.

Data structure for price lists is shown by the example of software for RDBMS MySQL

Table `prices`

`gid` integer is the group number (it must match the group number in the table "users")

`week_day` tinyint(1) is the day of the week as follows:

0 - Sunday

1 - Monday

2 - Tuesday

- 3 - Wednesday
- 4 - Thursday
- 5 - Friday
- 6 - Saturday
- 7 - holiday

`h0` double(8,6) is charge per hour from 00:00:00 until 00:59:59
`input0` double(8,6) is the charge for incoming traffic from 00:00:00 until 00:59:59
`output0` double(8,6) is the charge for outgoing traffic from 00:00:00 until 00:59:59
`h1` double(8,6) is the charge per hour from 01:00:00 until 01:59:59
`input1` double(8,6) is the charge for incoming traffic from 01:00:00 until 01:59:59
`output1` double(8,6) is the charge for outgoing traffic from 01:00:00 until 01:59:59
`h2` double(8,6) is the charge per hour from 02:00:00 until 02:59:59
`input2` double(8,6) is the charge for incoming traffic from 02:00:00 until 02:59:59
`output2` double(8,6) is the charge for outgoing traffic from 02:00:00 until 02:59:59
`h3` double(8,6) is the charge per hour from 03:00:00 until 03:59:59
`input3` double(8,6) is the charge for incoming traffic from 03:00:00 until 03:59:59
`output3` double(8,6) is the charge for outgoing traffic from 03:00:00 until 03:59:59
`h4` double(8,6) is the charge per hour from 04:00:00 until 04:59:59
`input4` double(8,6) is the charge for incoming traffic from 04:00:00 until 03:59:59
`output4` double(8,6) is the charge for outgoing traffic from 04:00:00 until 04:59:59
`h5` double(8,6) is the charge per hour from 05:00:00 until 05:59:59
`input5` double(8,6) is the charge for incoming traffic from 05:00:00 until 05:59:59
`output5` double(8,6) is the charge for outgoing traffic from 05:00:00 until 05:59:59
`h6` double(8,6) is the charge per hour from 06:00:00 until 06:59:59
`input6` double(8,6) is the charge for incoming traffic from 06:00:00 until 06:59:59
`output6` double(8,6) is the charge for outgoing traffic from 06:00:00 until 06:59:59
`h7` double(8,6) is the charge per hour from 07:00:00 until 07:59:59
`input7` double(8,6) is the charge for incoming traffic from 07:00:00 until 07:59:59
`output7` double(8,6) is the charge for outgoing traffic from 07:00:00 until 07:59:59
`h8` double(8,6) is the charge per hour from 08:00:00 until 08:59:59
`input8` double(8,6) is the charge for incoming traffic from 08:00:00 until 08:59:59
`output8` double(8,6) is the charge for outgoing traffic from 08:00:00 until 08:59:59
`h9` double(8,6) is the charge per hour from 09:00:00 until 09:59:59
`input9` double(8,6) is the charge for incoming traffic from 09:00:00 until 09:59:59
`output9` double(8,6) is the charge for outgoing traffic from 09:00:00 until 09:59:59
`h10` double(8,6) is the charge per hour from 10:00:00 until 10:59:59
`input10` double(8,6) is the charge for incoming traffic from 10:00:00 until 10:59:59
`output10` double(8,6) is the charge for outgoing traffic from 10:00:00 until 10:59:59
`h11` double(8,6) is the charge per hour from 11:00:00 until 11:59:59
`input11` double(8,6) is the charge for incoming traffic from 11:00:00 until 11:59:59
`output11` double(8,6) is the charge for outgoing traffic from 11:00:00 until 11:59:59
`h12` double(8,6) is the charge per hour from 12:00:00 until 12:59:59
`input12` double(8,6) is the charge for incoming traffic from 12:00:00 until 12:59:59
`output12` double(8,6) is the charge for outgoing traffic from 12:00:00 until 12:59:59
`h13` double(8,6) is the charge per hour from 13:00:00 until 13:59:59
`input13` double(8,6) is the charge for incoming traffic from 13:00:00 until 13:59:59
`output13` double(8,6) is the charge for outgoing traffic from 13:00:00 until 13:59:59
`h14` double(8,6) is the charge per hour from 14:00:00 until 14:59:59
`input14` double(8,6) is the charge for incoming traffic from 14:00:00 until 14:59:59

`output14` double(8,6) is the charge for outgoing traffic from 14:00:00 until 14:59:59
 `h15` double(8,6) is the charge per hour from 15:00:00 until 15:59:59
 `input15` double(8,6) is the charge for incoming traffic from 15:00:00 until 15:59:59
 `output15` double(8,6) is the charge for outgoing traffic from 15:00:00 until 15:59:59
 `h16` double(8,6) is the charge per hour from 16:00:00 until 16:59:59
 `input16` double(8,6) is the charge for incoming traffic from 16:00:00 until 16:59:59
 `output16` double(8,6) is the charge for outgoing traffic from 16:00:00 until 16:59:59
 `h17` double(8,6) is the charge per hour from 17:00:00 until 17:59:59
 `input17` double(8,6) is the charge for incoming traffic from 17:00:00 until 17:59:59
 `output17` double(8,6) is the charge for outgoing traffic from 17:00:00 until 17:59:59
 `h18` double(8,6) is the charge per hour from 18:00:00 until 18:59:59
 `input18` double(8,6) is the charge for incoming traffic from 18:00:00 until 18:59:59
 `output18` double(8,6) is the charge for outgoing traffic from 18:00:00 until 18:59:59
 `h19` double(8,6) is the charge per hour from 19:00:00 until 19:59:59
 `input19` double(8,6) is the charge for incoming traffic from 19:00:00 until 19:59:59
 `output19` double(8,6) is the charge for outgoing traffic from 19:00:00 until 19:59:59
 `h20` double(8,6) is the charge per hour from 20:00:00 until 20:59:59
 `input20` double(8,6) is the charge for incoming traffic from 20:00:00 until 20:59:59
 `output20` double(8,6) is the charge for outgoing traffic from 20:00:00 until 20:59:59
 `h21` double(8,6) is the charge per hour from 21:00:00 until 21:59:59
 `input21` double(8,6) is the charge for incoming traffic from 21:00:00 until 21:59:59
 `output21` double(8,6) is the charge for outgoing traffic from 21:00:00 until 21:59:59
 `h22` double(8,6) is the charge per hour from 22:00:00 until 22:59:59
 `input22` double(8,6) is the charge for incoming traffic from 22:00:00 until 22:59:59
 `output22` double(8,6) is the charge for outgoing traffic from 22:00:00 until 22:59:59
 `h23` double(8,6) is the charge per hour from 23:00:00 until 23:59:59
 `input23` double(8,6) is the charge for incoming traffic from 23:00:00 until 23:59:59
 `output23` double(8,6) is the charge for outgoing traffic from 23:00:00 до 23:59:59

Below is the tariffs data structure shown by the example of the description of the table "packets" version 2.0 for RDBMS MySQL.

Table `packets`

 `num` Integer is the packet number (*)
 `packet` varchar(128) is the packet's name (*)
 `prefix` varchar(4) is the packet's prefix
 `gid` integer is the group number
 `deposit` double is the initial account value (*)
 `credit` double is the initial credit value (*)
 `tos` tinyint(1) is what is to be charged:
 0 means do not charge
 1 means charge for time
 2 means charge for traffic
 3 means charge for time + traffic
 `do_with_tos` tinyint(1) - draw money from the deposit and check whether the account is in funds:
 0 - no, only enter in the statistics table (used only when working with monetary limits)

1 – yes and enter in the statistics table

`direction` tinyint(1) – when charging for traffic count traffic:

- 0 – do not count
- 1 - outgoing
- 2 - incoming
- 3 - total
- 4 - bigger (for the entire connection)
- 5 - lesser (for the entire connection)

`fixed` tinyint(1) – draw fixed sum:

- 0 – do not draw
- 1 – once per day if there was a connection
- 2 – for each day including the previous day
- 3 – with every connection

`fixed_cost` double - amount of money to be drawn

`activated` tinyint(1) - initial value of 'activated' (*)

`activation_time` int(10) - activation time in seconds

`blocked` tinyint(1) - if the value is 1, then all packet is blocked

`total_time_limit` int(10) - total time limit (**)

`month_time_limit` int(10) - month time limit (**)

`week_time_limit` int(10) - week time limit (**)

`day_time_limit` int(10) - day time limit (**)

`total_traffic_limit` bigint(15) - total traffic limit (**)

`month_traffic_limit` bigint(15) - month traffic limit (**)

`week_traffic_limit` bigint(15) - week traffic limit (**)

`day_traffic_limit` bigint(15) - day traffic limit (**)

`total_money_limit` double - total money limit (**)

`month_money_limit` double - month money limit (**)

`week_money_limit` double - week money limit (**)

`day_money_limit` double - day money limit (**)

`login_time` varchar(254) - special format string to define the allowed time intervals for the connection

`huntgroup_name` varchar(64) - IP group name
(etc/raddb/huntgroups)

`simultaneous_use` smallint(5) - number of simultaneously connected subscribers

`port_limit` smallint(5) - port limit for one login

`session_timeout` int(10) - session timeout

`idle_timeout` int(10) - idle timeout

`allowed_prefixes` varchar(64) - allowed prefixes for the change of the tariff

`framed_ip` varchar(16) - framed IP address (recommended with '+' symbol at the end, e. g. 192.168.100.0+)

`framed_mask` varchar(16) - framed mask for the packet (e. g. 255.255.255.255)

`no_pass` tinyint(1) - if the value is 1, then grant access to all the subscribers of this packet without password

`no_acct` tinyint(1) - if the value is 1, then do not track statistics for the subscribers of this packet (money will not be counted either)

`allow_callback` tinyint(1) - allow callback for this packet

`other_params` varchar(254) - text field for the values transferred to the access server
format: parameter = value, parameter = value

`create_system_user` tinyint(1) - create system user when creating user in the database
(*)

Setting of the tariffs visibility for operators/resellers.

Login as provider's operator. Choose menu item "Providers->Resellers".

The screenshot shows a web application interface for managing resellers. The browser window is titled "[NETSHe-0.1] - Opera". The address bar shows the URL: `http://localhost:5556/suite/provider/index.php?action=sys.ProvResellers.table_view&page=0`. The interface includes a navigation menu with options like "DialIn", "Reports", "Provider", and "Data". A table titled "[Resellers]" displays a list of resellers with columns for #, Title, Companies Name, CEO, CFO, CAO, CTO, Responsible person, Document, and Option. A help dialog box is open, showing icons for New, Edit, Find, Delete, Lock, and Operators. The status bar at the bottom shows the time as 13:01:34.

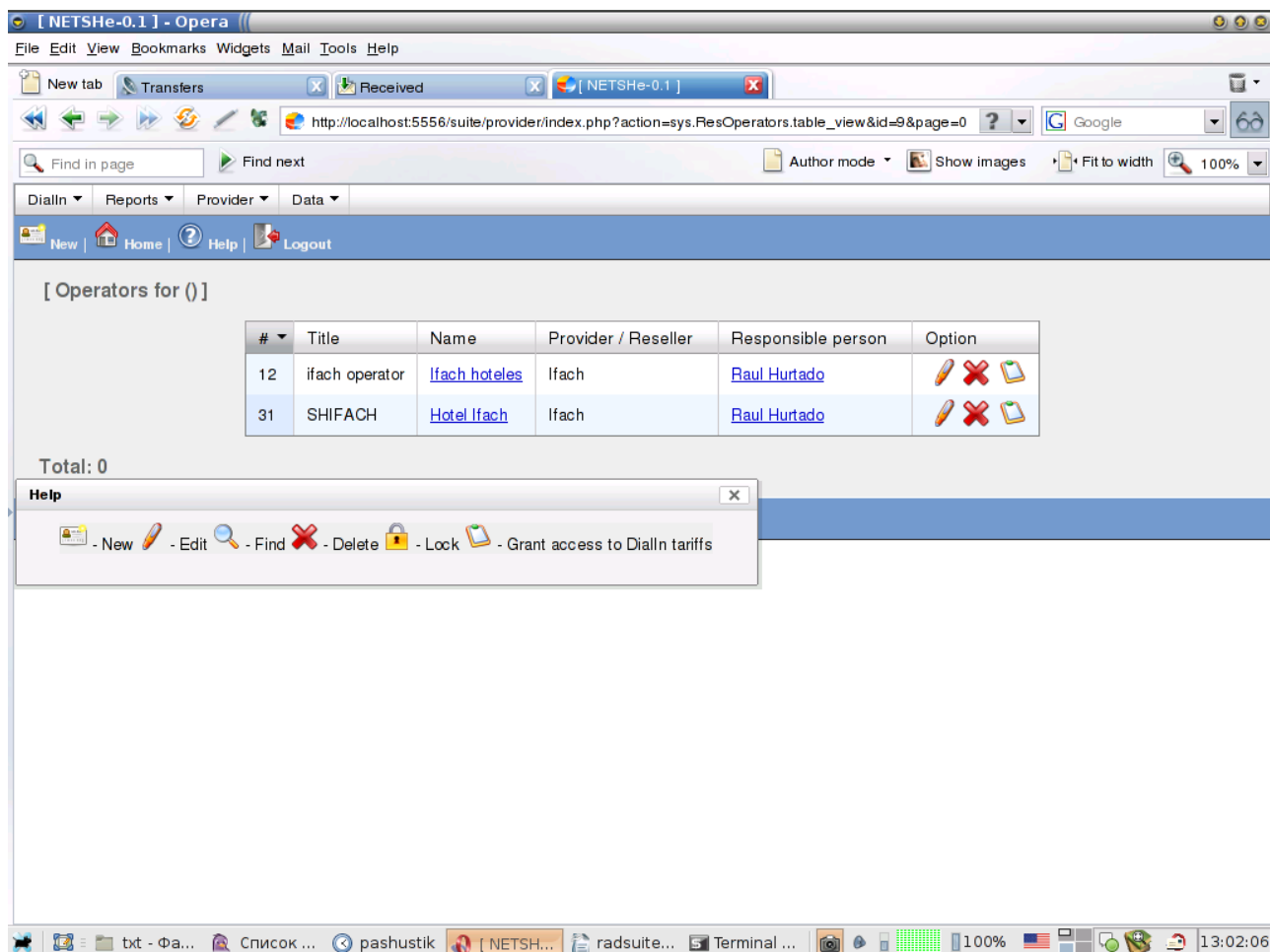
#	Title	Companies Name	CEO	CFO	CAO	CTO	Responsible person	Document	Option
4	Tryp Florazar	OPTIMIZA	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Raul Hurtado		
5	Meliá Valencia Palace	Palace hotel	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Raul Hurtado		
6	Meliá Inglés Boutique	Ingles hotel	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Raul Hurtado		
7	SH Hoteles	Ifach hotel	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Raul Hurtado		
9	Ifach	Ifach hotel	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Raul Hurtado		
16	Villa Gadea	Hotel Villa Gadea	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Raul Hurtado		
17	AlteaHills	Hotel Altea Hills	Josep Pocalles	Josep Pocalles	Josep Pocalles	Josep Pocalles	Raul Hurtado		

Total: 7

Help

- New - Edit - Find - Delete - Lock - Operators

Choose the list of operators of the corresponding reseller in the column "Options".



And go to the tariffs access control for a specific operator.

Note: You can create a new operator.

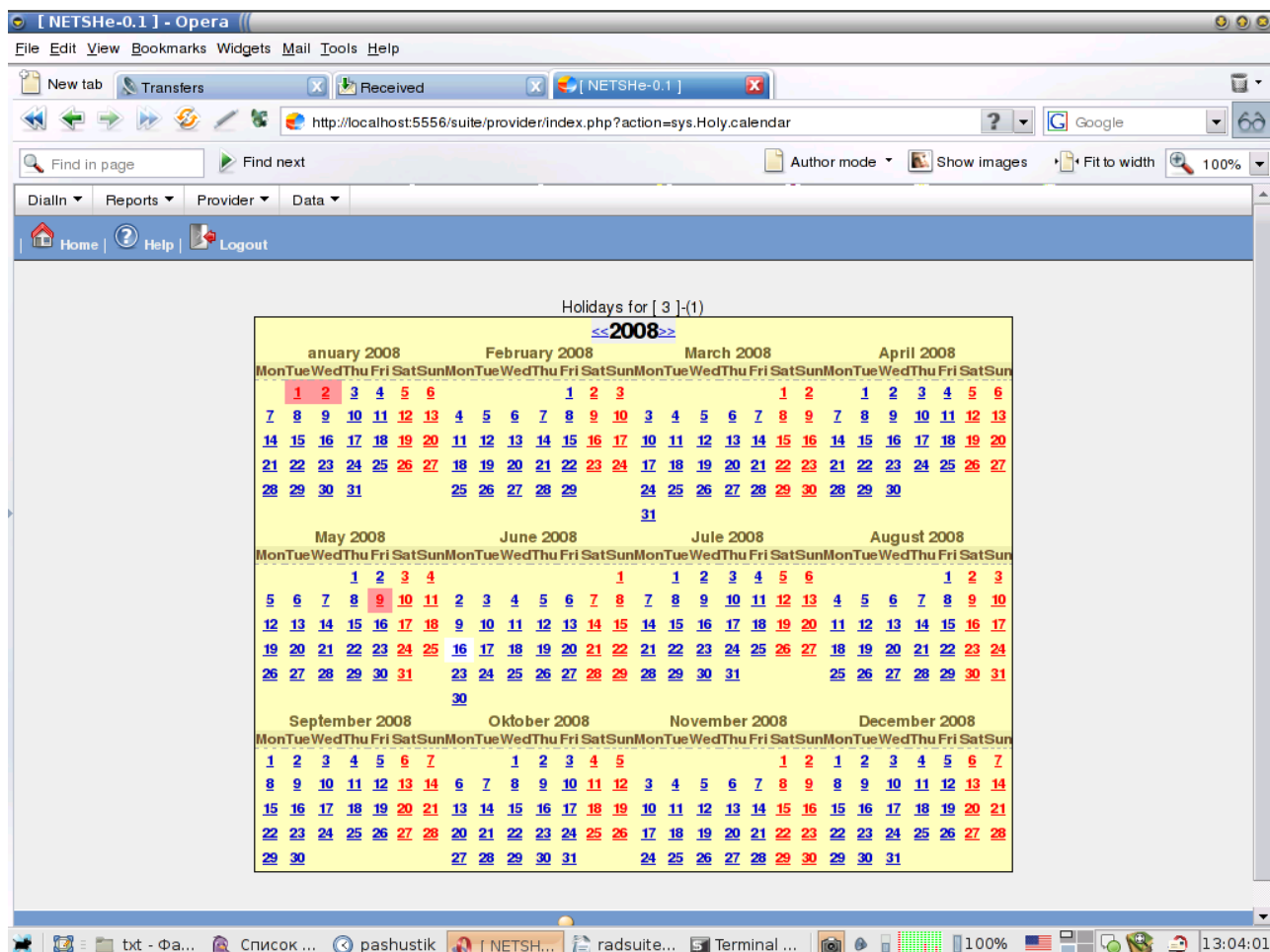
Then by mouse click in the field “Options” change the accessibility of the tariff for the given operator.

Note: In the general list of tariffs you will see only the tariffs of the given provider.

By making specific tariffs accessible for the operator you thereby allow viewing statistics for the given subscriber/tariff, creating/editing subscribers and modifying the tariff¹.

After the definition of the tariffs and their accessibility to the system operators you can define the holidays (“special” days). To do this choose the menu item “Dialin->Holidays”.

¹ Exact data modification rights are determined by the additional system operator assignments



Creation/cancellation of holidays is performed by simple mouse click on the right date.
 Note: Holidays are to be defined only when using tariffs with billing type “time”, “traffic (without gradations)” and “time+traffic”.

Holidays data structure for RDBMS MySQL

Table `holidays`

- `holiday_date` varchar(5) – date of holiday in MM-DD format
- `comment` varchar(64) – description of holiday (*)

After the definition of the tariffs and the holidays you can enter the subscribers.

Creating and editing subscribers.

Creation of subscribers in the system can be performed by the following two methods:

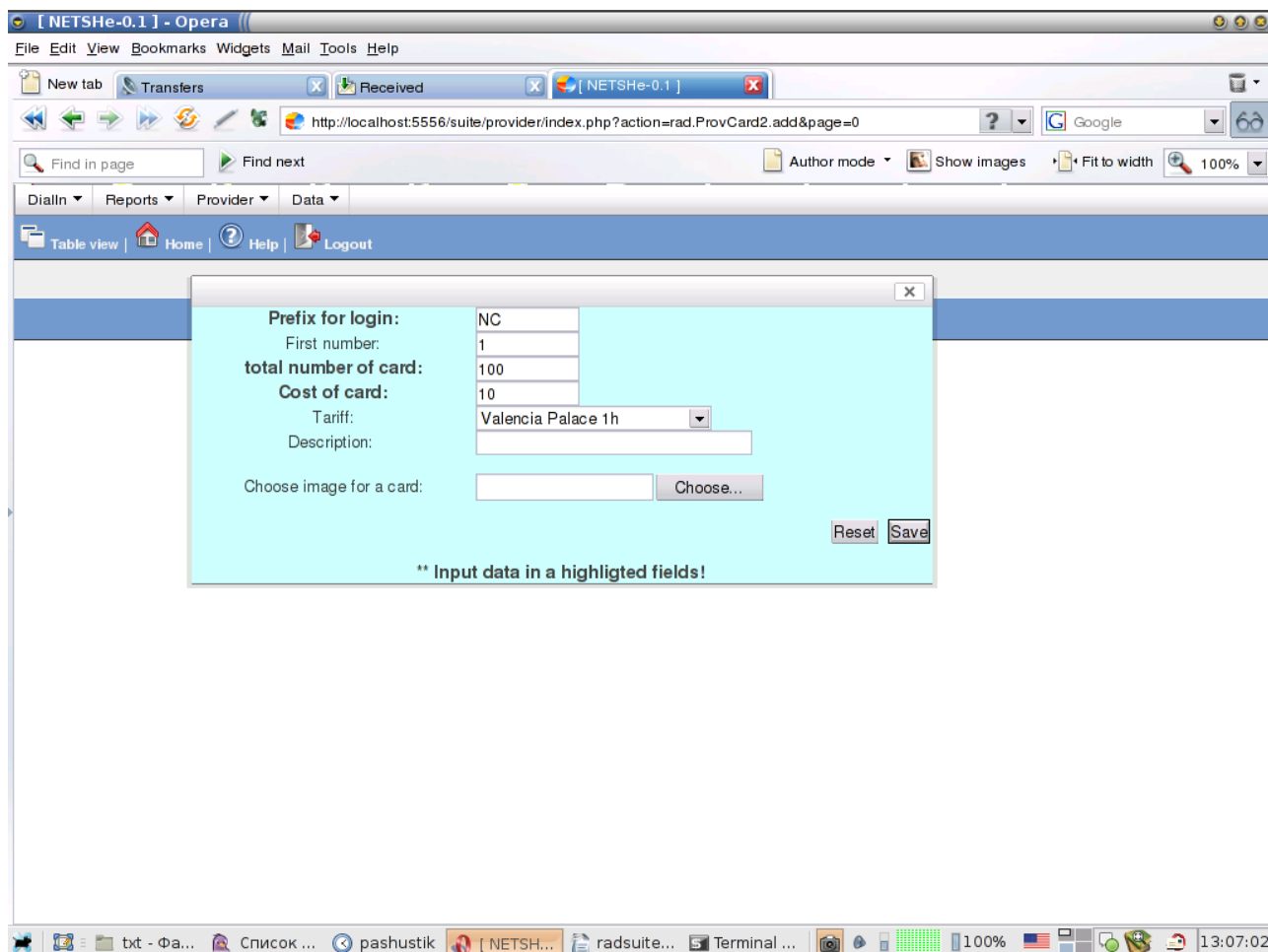
- By means of creation of prepaid cards, where a card is referred to a certain tariff and contains all the necessary connection data (login/password, etc.). In this case the subscribers are created as a group and logins/passwords are generated automatically. All cards (subscribers) within the series have identical settings. The system has the necessary tools for creation of individual images, csv-file generation with logins/passwords and card printing (pdf-files are generated) and login/password sending to the subscriber as an SMS (clicatell service is supported but other services can be used as well);

- By means of entering new subscribers. In this case subscribers are created one by one.

Let us review both methods beginning with the prepaid cards method. To work with series of prepaid cards choose the menu item “Dialin->Cards”.

#	Image	Cost of card	Creation date	Users	Tariff	Description	Option
55		12.00	1970-01-01 00:00:00	100	Valencia Palace 24h	Valencia Palace 24 horas	
61		6.50	1970-01-01 00:00:00	100	Valencia Palace 1h	ValPalace1h203rs	
68		12.00	1970-01-01 00:00:00	1	Valencia Palace 24h	fixproblem	
69		6.50	1970-01-01 00:00:00	100	Valencia Palace 1h	Florazar 1h	

The table contains card image, specifies card price, number of cards in the series, tariff, etc. The column “Options” shows which operations can be performed with the card series (locking/deletion/printing to csv-file or pdf/file, etc.). A special form is used for creation of a new card series:



In order to generate a series of cards choose the tariff, enter login prefix (initial login symbols, which are common for all the cards in the series), first card number in the series, number of cards in the series and cost of the card.

The newly created cards appear in the general list of subscribers and deletion of the card series leads to deletion of all the cards (subscribers) in that series.

To manage subscribers (including creation of new subscribers one by one) choose the menu item "Dialin->Subscribers".

[NETSHe-0.1] - Opera

File Edit View Bookmarks Widgets Mail Tools Help

Transfers Received [NETSHe-0.1]


































http://localhost:5556/suite/provider/index.php?action=rad.ProvU ser2.table_view&page=0

Find in page Find next Author mode Show images Fit to width 100%

DialIn Reports Provider Data

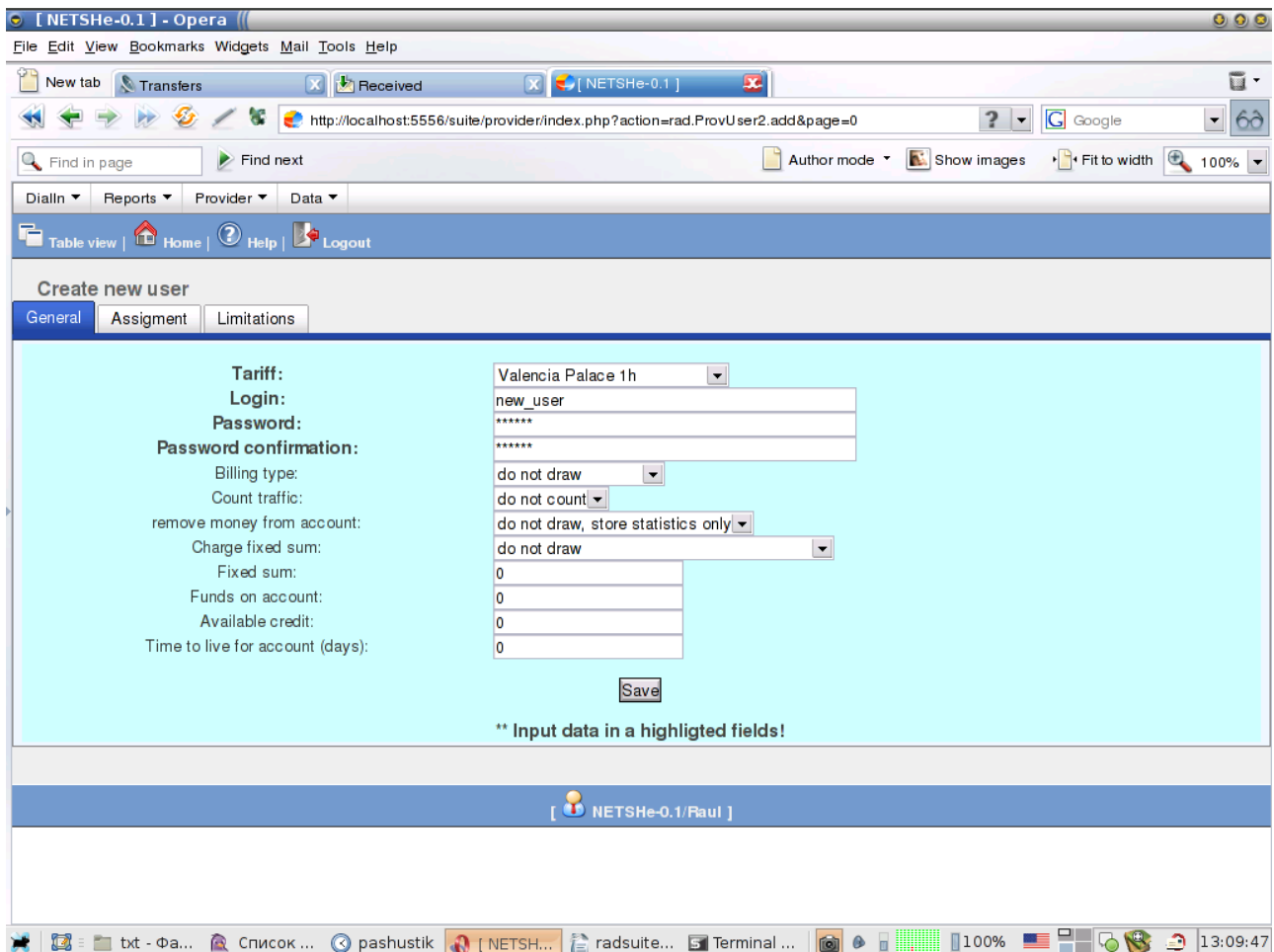
New Find Next End Home Help Logout

[Users list]

#	Login	Tariff	Funds on account	Available credit	Expired	Last connection	Used time (hh:mm:ss)	Used traffic (MB)	Used money	Option
100	eko34307	1 hour - SH1	6.00	0.00	2007-07-05 16:53:49	1970-01-01 00:00:00	0:0:0	0	0.00	  
101	eko34308	1 hour - SH1	6.00	0.00	2007-07-06 12:28:52	1970-01-01 00:00:00	0:0:0	0	0.00	  
102	eko34309	1 hour - SH1	6.00	0.00	2007-07-06 12:30:50	1970-01-01 00:00:00	0:0:0	0	0.00	  
103	eko34310	1 hour - SH1	6.00	0.00	2007-07-06 12:32:34	1970-01-01 00:00:00	0:0:0	0	0.00	  
104	eko34311	1 hour - SH1	6.00	0.00	2007-07-07 14:02:20	1970-01-01 00:00:00	0:0:0	0	0.00	  
105	eko34312	1 hour - SH1	6.00	0.00	2007-07-12 16:02:26	1970-01-01 00:00:00	0:0:0	0	0.00	  
106	eko34313	1 hour - SH1	6.00	0.00	2007-07-14 12:28:34	1970-01-01 00:00:00	0:0:0	0	0.00	  
107	eko34314	1 hour - SH1	6.00	0.00	2007-08-22 10:11:57	1970-01-01 00:00:00	0:0:0	0	0.00	  
108	eko34315	1 hour - SH1	6.00	0.00	2007-08-22 10:12:08	1970-01-01 00:00:00	0:0:0	0	0.00	  
109	eko34316	1 hour - SH1	6.00	0.00	2007-08-23 10:19:41	1970-01-01 00:00:00	0:0:0	0	0.00	  
110	eko34317	1 hour - SH1	6.00	0.00	2007-08-23 11:22:32	1970-01-01 00:00:00	0:0:0	0	0.00	  

txt - Фа... Список ... pashustik [NETSH... radsuite... Terminal ... 100% (null) 13:08:36

This table contains such information as subscriber number, login, tariff, sum on account, available credit, some statistical data and buttons for subscriber management. Subscriber editing/creation form is very similar the tariff editing form.



Below is the subscriber data structure in RDBMS MySQL

Table `Subscribers`

```

`user` varchar(64) – subscriber login
    * must be of lowercase letters of the English alphabet
    * can contain numbers as well as symbols `` and ``
    * must not begin with a number
`passwd` varchar(254) - password
`crypt_method` tinyint(1) – type of password encryption
    0 – not encrypted password
    1 - encrypted Crypt
    2 - encrypted MD5
    3 - encrypted SHA1
`uid` integer – unique subscriber ID
`gid` integer – group (packet) ID by default
`deposit` double - deposit
`credit` double - credit
    If Deposit + Credit <= 0 at packet.tos > 0 then access is denied
`fio` varchar(128) – subscriber name (*)
`phone` varchar(128) – contact phone number (*)
`address` varchar(128) - address (*)
  
```


``prim` varchar(254) - note (*)`
``add_date` date(YYYY-MM-DD) – time of subscriber creation/activation`
 * If during subscriber creation the field has 0000-00-00, then it will be set to the current date.
 * If the field value supersedes the current date, then access for the subscriber will be denied until the defined date.
``blocked` tinyint(1) – if the value is 1, then access is denied`
``activated` tinyint(1) - if the value is 0, then during logging in will be set to 1 and if`
`packets.activation_time` is more than 0, then the expired field will be corrected. `expired =`
`current date + number of seconds in packets.activation_time.`
``expired` date(YYYY-MM-DD) – login expiry date.`
 * If the value is 0, then ignored.
``total_time` int(10) – time counter in seconds`
 * Used for comparison with the field `packets.total_time_limit` and as soon as its value exceeds the value of the latter the access is denied.
``total_traffic` bigint(15) – traffic counter in bytes`
 * Used for comparison with the field `packets.total_traffic_limit` and as soon as its value exceeds the value of the latter the access is denied.
``total_money` double – spent money counter.`
 * Used for comparison with the field `packets.total_money_limit` and as soon as its value exceeds the value of the latter the access is denied
``last_connection` date(YYYY-MM-DD) – last connection date`
 * Used for charging the fixed sum.
``framed_ip` varchar(16) - assigned IP`
``framed_mask` varchar(16) – assigned mask`
``callback_number` varchar(64) – callback number if allowed in packets.allow_callback`
 * If the field is empty and `packets.allow_callback` for this group is set to 1, then the subscriber will be invited to enter any number.

Statistics of subscriber sessions can be viewed by clicking the buttons in the column "Options".

[Session history - Ordered by 'PLC648', contains "]

User name / Login	Session start date	Session end date	Call from	Server:Port	Address	Session Duration (hh:mm:ss)	Sent (KB)	Received (KB)	Billed	Termination Cause
PLC648	2008-03-06 18:44:45	2008-03-06 19:44:46	00-1C-BF-4A-8E-17	0.0.0.0:28	192.168.182.121	1:0:1	478	7230	0	Session-Timeout

Total: records found-1, time (hh:mm:ss)-1:0:1, sent KB-478, received KB-7230, sum-0.000000

[NETSHe-0.1/Raul]

Below is the sessions data structure in RDBMS MySQL

Table `actions`

- `user` varchar(64) – subscriber login
- `gid` integer – group the subscriber is currently in
- `id` varchar(32) – connection ID
- `unique_id` varchar(64) – unique connection ID
- `time_on` int(12) – connection duration in seconds
- `start_time` datetime(YYYY-MM-DD HH:MM:SS) – connection start date and time
- `stop_time` datetime(YYYY-MM-DD HH:MM:SS) – connection stop date and time
- `in_bytes` bigint(15) – received bytes
- `out_bytes` bigint(15) – transferred bytes
- `ip` varchar(15) – subscriber IP
- `server` varchar(15) – access server IP
- `client_ip` varchar(15) – real access server IP
- `port` int(12) – access server port
- `call_to` varchar(64) – phone number, which has been called
- `call_from` varchar(64) – subscriber phone number
- `connect_info` varchar(254) – connection information
- `protocol` varchar(32) – used protocol
- `terminate_cause` varchar(64) – connection termination cause

`hour_in_bytes` bigint(15) – bytes received for the last (current) hour
`hour_out_bytes` bigint(15) - bytes transferred for the last (current) hour
`hour_traffic_money` double – charge incurred for the traffic of the last (current) hour
`last_change` int(10) – timestamp of the last record update
`before_billing` double – account status at connection start
`billing_minus` double – amount of money drawn for the connection

Statistical data and reports

The following statistical data and reports are available in the system (main items listed only):

- ▣ Viewing of online subscribers.
- ▣ Viewing of all sessions of the subscriber.
- ▣ Reports on the traffic, time and incurred charges for the services.
- ▣ Reports on the subscribers with expired lifetime.

Annex 1

Abbreviations used in SQL requests in nibs.conf

Request	Description	Equivalent
-----	-----	-----
%a	Protocol (SLIP/PPP)	%{Framed-Protocol}
%c	Callback-Number	%{Callback-Number}
%d	Request date (DD)	
%f	Framed IP address	%{Framed-IP-Address}
%i	Calling Station ID	%{Calling-Station-Id}
%l	Request timestamp	
%m	Request month (MM)	
%n	NAS IP address	%{NAS-IP-Address}
%p	Port number	%{NAS-Port-Id} %{NAS-Port}
%s	Speed (PW_CONNECT_INFO)	%{Connect-Info}
%t	Request in ctime format	
%u	User name	%{User-Name}
%A	radacct_dir	%{config:radacctdir}
%C	clientname	
%D	Request date (YYYYMMDD)	
%H	Request hour	
%L	radlog_dir	%{config:logdir}
%M	MTU	%{Framed-MTU}
%R	radius_dir	%{config:raddbdir}
%S	Request timestamp in SQL format	
%T	Request timestamp in database format	
%U	Stripped User name	%{Stripped-User-Name}
%V	Request-Authenticator (Verified/None)	
%Y	Request year (YYYY)	
%Z	All request attributes except password (must have a big buffer)	

Annex 2. Setting of individual parameters

Stale sessions

There are two parameters in nibs.conf file:

```
deletestalesessions_by_checkrad = yes
deletestalesessions_by_alive = yes
```

The first parameter allows to check for double connections and determine stale sessions by means of checkrad script invocation. This parameter is applicable only for hardware access servers (NAS). Correct tuning of clients.conf is required and especially the following parameters:

```
nastype    = server type
login      = login for connection
password   = password
```

Below is the list of supported servers taken from the original FreeRadius documentation.

Type	Manufacturer	Access method	Required items	Password required
====	=====	=====	=====	=====
ascend	Lucent	SNMP	SNMP	No
bay	Nortel	finger	finger command	No
cisco	Cisco	SNMP	SNMP	Optional [1]
computone	Computone	finger	finger command	No
cvx	Nortel	SNMP	SNMP	No
digitro	Digitro	rusers	rusers command	No
livingston	Livingston	SNMP	SNMP	No [2]
max40xx	Lucent	finger	finger command	No
netserver	USR/3com	telnet	CPAN Net::Telnet	Yes
pathras	Cyclades	telnet	CPAN Net::Telnet	Yes
patton	Patton	SNMP	SNMP	No
portslave	?	finger	finger command	No
pr3000	Cyclades	SNMP	snmpwalk command	No
pr4000	Cyclades	SNMP	snmpwalk command	No
tc	USR/3com	telnet	CPAN Net::Telnet	Yes
usrhyper	USR/3com	SNMP	SNMP	No [3]
versanet	VersaNet	SNMP	SNMP	No

[1] The following settings are to be made in the configuration file: login = SNMP, password = required community.

[2] Needs at least ComOS 3.5, SNMP enabled.

[3] Set "Reported Port Density" to 256 (default)

The second parameter allows to check for the double connection and determine stale sessions by means of ALIVE packet arrival time calculation. It is applicable for any types of NAS but requires appropriate ALIVE configuration.

This parameter operates as follows:

The value of last_change is taken from the table 'actions' and the value of Acct-Interim-Interval is taken from users.other_params. Stale session is calculated by the following formula: $((\text{now}() - \text{last_change}) \geq (\text{Acct-Interim-Interval} + \text{alive_max_interval}))$, i. e. if within alive_max_interval seconds the expected ALIVE does not arrive the session is considered "stale".

Note: Session is considered stale if the connection is established from the port and the access server, which are already specified in the table 'actions' as "Online". Such sessions are closed automatically.

Annex 3. Description of external subscriber/service management interface

External management API

1. session_start function. Should be first in a sequence. Parameters are - username, md5(password), key, message

Key is a randomly generated sequence. Msg is a message that contains a description of an operation, payment method, etc. (Like 'PayPal payment')

Example of invocation: `http://url/vpi/mux.php?`

`action=session_start&username=payment_gw&password=XXX&key=YOUR_KEY&message=YYY`

Return value - xml-file with session_id (string length 128bit) or with fail in case of error

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response>
```

```
<response_code>ok</response_code>
```

```
<session>34VDNI123AAWQERWVC56</session>
```

```
</response>
```

or

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>fail</response_code></response>
```

2. session_end function. Should be last in a sequence. Parameter are - sequence_id as F(session)

All sessions have time limits. It increases error_counter for client, when session was not ended

with session_end call. And may cause client blocking.

Example of invocation: `http://url/vpi/mux.php?action=session_end&sequence_id=YYY`

Return values - xml-file with ok or fail.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>ok</response_code></response>
```

or

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>fail</response_code></response>
```

3. get_provider_list function. Parameters are - sequence_id as F(session), service as string

Example of invocation: `http://url/vpi/mux.php?`

`action=get_provider_list&sequence_id=YYY&service=rad`

Return value - xml-file with a list of providers or fail in a case of error.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response>
```

```
<response_code>ok</response_code>
```

```
<providers>
```

```
<provider>
```

```
<id>1</id>
```

```

    <name>EkoPLC</name>
  </provider>
  <provider>
    <id>2</id>
    <name>Another Provider</name>
  </provider>
</providers>
</response>
or
<?xml version="1.0" encoding="UTF-8" ?>
<response><response_code>fail</response_code></response>

```

4. get_tariff_list function. Parameters are - sequence_id as F(session), service as string, provider_id as number

Example of invocation: [http://url/vpi/mux.php?](http://url/vpi/mux.php?action=get_tariff_list&sequence_id=YYY&service=rad&provider_id=15)

action=get_tariff_list&sequence_id=YYY&service=rad&provider_id=15

Return value - xml-file with a list of tariffs or fail in a case of error.

```

<?xml version="1.0" encoding="UTF-8" ?>

```

```

<response>
<response_code>ok</response_code>
<tariffs>
  <tariff>
    <id>1</id>
    <name>Best time</name>
    <subject>Time</subject>
    <cost>2</cost>
    <currency>EUR</currency>
  </tariff>
  <tariff>
    <id>2</id>
    <name>Another tariff</name>
    <subject>Traffic</subject>
    <cost>2</cost>
    <currency>USD</currency>
  </tariff>
  <tariff>
    <id>2</id>
    <name>Hotel - 24 hours</name>
    <subject>Fixed</subject>
    <cost>12</cost>
    <currency>EUR</currency>
  </tariff>
</tariffs>
</response>

```

```

or
<?xml version="1.0" encoding="UTF-8" ?>
<response><response_code>fail</response_code></response>

```


5. disable_user function. Parameters are - sequence_id as F(session), service as string, user_id as number, cause as string

Example of invocation: http://url/vpi/mux.php?

action=disable_user&sequence_id=YYY&service=rad&user_id=15,cause=XXX

Return value - xml-file with an ok or fail in a case of error.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>ok</response_code></response>
```

or

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>fail</response_code></response>
```

6. enable_user function. Parameters are - sequence_id as F(session), service as string, user_id as number, cause as string

Example of invocation: http://url/vpi/mux.php?

action=enable_user&sequence_id=YYY&service=rad&user_id=15,cause=XXX

Return value - xml-file with an ok or fail in a case of error.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>ok</response_code></response>
```

or

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>fail</response_code></response>
```

7. create_user function. Parameters are - sequence_id as F(session), service as string, tariff_id as number, cause as string

Cause is a message, that contains cause of user creation: action from payment gateway, etc.

Example of invocation: http://url/vpi/mux.php?

action=create_user&sequence_id=YYY&service=rad&tariff_id=15,cause=XXX

Return value - xml-file with an ok, username, password, user_id and rechargable flag ('yes' or 'no')

or fail in a case of error.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response>
```

```
    <response_code>ok</response_code>
```

```
    <username>TEST</username>
```

```
    <password>123</password>
```

```
    <user_id>15008</user_id>
```

```
    <can_be_recharged>yes</can_be_recharged>
```

```
</response>
```

Created user has zero account balance and has disabled.

or

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>fail</response_code></response>
```

8. proceed_payment function. Parameters are - sequence_id as F(session), service as string, user_id as number, sum as number, currency as string, doc_id as string, cause as string, hash as string

Cause should contain type of payment (PayPal, Credit card - VISA, etc..)

Currency - currency sign for payment, because suite is a multicurrency application.
Doc_id is a unique transaction id from payment gateway processing software.

Example of invocation: `http://url/vpi/mux.php?`

`action=proceed_payment&sequence_id=YYY&service=rad&user_id=15&sum=10¤cy=Euro&doc_id=123&cause=XXX&hash=ZZZ`

where hash is md5(concatenation of values of all parameters from left to right)

Return value - xml-file with an ok and amount or fail in a case of error.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response>
```

```
<response_code>ok</response_code>
```

```
<amount>12</amount>
```

```
</response>
```

or

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>fail</response_code></response>
```

9. `get_user_id` function. Parameters are - `sequence_id` as F(session) as string, `service` as string ('rad' for example), `uname` as string (username), `passwd` as string (users password).

Example of invocation: `http://url/vpi/mux.php?`

`action=get_user_id&sequence_id=XXX&service=rad&uname=test1&passwd=abra`

Return value - xml-file with on ok, `user_id` and rechargable flag or fail in a case of error.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response>
```

```
<response_code>ok</response_code>
```

```
<user_id>15009</user_id>
```

```
<can_be_recharged>no</can_be_recharged>
```

```
</response>
```

or

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response><response_code>fail</response_code></response>
```

10. `get_tariff_list_enabled` function. Parameters are - `sequence_id` as F(session), `service` as string

Example of invocation: `http://url/vpi/mux.php?`

`action=get_tariff_list_enabled&sequence_id=YYY&service=rad`

Return value - xml-file with a list of tariffs, enabled for this gateway's username/password or fail in a case of error.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<response>
```

```
<response_code>ok</response_code>
```

```
<tariffs>
```

```
<tariff>
```

```
<id>1</id>
```

```
<name>Best time</name>
```

```
<subject>Time</subject>
```

```
<cost>2</cost>
```

```

        <currency>EUR</currency>
    </tariff>
    <tariff>
        <id>2</id>
        <name>Another tariff</name>
        <subject>Traffic</subject>
        <cost>2</cost>
        <currency>USD</currency>
    </tariff>
    <tariff>
        <id>2</id>
        <name>Hotel - 24 hours</name>
        <subject>Fixed</subject>
        <cost>12</cost>
        <currency>EUR</currency>
    </tariff>
</tariffs>
</response>
or
<?xml version="1.0" encoding="UTF-8" ?>
<response><response_code>fail</response_code></response>

```

All invocation is performed by GET-method.

F(session) is

md5(YOUR_KEY+SESSION_ID) for step #1 (First call after session_start) and
md5(F(n-1)) for step #n